Gestión automática del correo electrónico usando técnicas de procesamiento de lenguaje natural: el caso de Labdoo

Julián Arenas Guerrero

GRADO EN INGENIERÍA INFORMÁTICA. FACULTAD DE INFORMÁTICA
UNIVERSIDAD COMPLUTENSE DE MADRID

Trabajo Fin de Grado en Ingeniería Informática

Fecha 19 de septiembre de 2019

Director:

Jorge J. Gómez Sanz

Resumen en castellano

Actualmente, el uso del correo electrónico sigue estando muy extendido en la comunicación y coordinación de las diferentes organizaciones. Esto aumenta la cantidad de datos que los/as usuarios/as tienen que procesar, provocando un gran nivel de estrés y ansiedad que disminuye la eficiencia. El problema es aún más pronunciado en el caso de las ONGs, que normalmente cuentan con escasos recursos económicos y humanos.

Con vistas a lo anterior, en este trabajo se ha realizado una prueba de concepto con el objetivo de mejorar la gestión del correo eletrónico y la automatización de tareas. Para ello, se ha trabajado con el caso real de la ONG *Labdoo*. Los resultados obtenidos son prometedores, y muestran que el 85 % de los correos pueden tratarse de manera automática, con una tasa de acierto muy elevada. Se espera que este proyecto contribuya de manera notable a la mejora de la gestión del correo electrónico en organizaciones con escasos recursos.

Palabras clave

Sobrecarga de correo electrónico, Gestión de flujo de trabajo, Correo electrónico, Clasificación de correos, Aprendizaje automático, PLN, ONG.

Abstract

Currently, the use of email is still widespread in the communication and coordination of different organizations. This increases the amount of data that users have to process, causing high levels of stress and anxiety that decrease efficiency. The problem is even more pronounced in the case of NGOs, which usually have limited economic and human resources.

In view of the above, in this work a proof of concept has been carried out in order to improve the management of email and the automation of tasks. With this purpose, we have worked with the real case of the NGO Labdoo. The obtained results are promising, and show that 85 % of emails can be treated automatically, with a very high success rate. This project is expected to contribute significantly to the improvement of email management in organizations with limited resources.

Keywords

Email overload, Workflow management, Email, Email classification, Machine learning, NLP, NGO.

Índice general

Agradecimientos

A mis padres Julián y Carmen, a mi hermana Paloma y a mi hermano Alberto.

Capítulo 1

Introducción

1.1. Presentación y motivación

Actualmente, a pesar de la disponibilidad de redes sociales, gestores de flujo de trabajo, o la Web 2.0, el **correo electrónico** sigue teniendo un peso importante en la comunicación a todos los niveles. De hecho, su uso medio durante la jornada laboral se cifra en 2.6 horas [12]. En muchas organizaciones, es un hábitat [14] donde se recibe, gestiona y delega el trabajo, y que se utiliza para diversidad de tareas, como gestión de la información, coordinación y colaboración, o para la gestión de contactos.

Este uso extendido del correo electrónico aumenta la cantidad de datos que los/as usuarios/as tienen que procesar, provocando un gran nivel de estrés y ansiedad [13] [20] que hace perder eficacia a la hora de utilizarlo. El concepto *email overload* se define como la sensación de agobio por el flujo constante de mensajes que aparecen en la bandeja de entrada, y la incapacidad para manejar el alto volumen de estos eficientemente [20]. Esta sensación se ve intensificada por la continua interrupción de tareas, el flujo de trabajo y de pensamiento [16] [18].

Ante este problema, una de las soluciones más directas para los/as usuarios/as consiste

en usar diferentes técnicas para abordar más rápidamente los mensajes, tales como priorizar, agrupar o filtrar emails. De hecho, recientemente se han definido nuevos enfoques orientados a aumentar la productividad, como *inbox zero*, que propone una serie de pautas para mejorar la gestión del correo electrónico por parte de los/as usuarios/as [7].

Avances en los campos de la Inteligencia Artificial y la Recuperación de Información son susceptibles de ayudar en la gestión del correo. Este tipo de tecnologías ya se han utilizado, por ejemplo, para la priorización de tareas o para la clasificación de emails [15] [21]. No obstante, no existen aún herramientas disponibles que implementen estas técnicas junto a funcionalidad del correo como la traducción o filtrado. Además, la gran variedad de tareas para las que se usa el correo hace que desarrollar soluciones generales sea una tarea compleja y no trivial, puesto que las necesidades dependen del tipo de usuario/a y de organización.

En este trabajo, nos centramos en la automatización de tareas y la mejora de la gestión del correo eletrónico en el caso concreto de las **ONGs**. La característica principal de este tipo de organizaciones, por su carácter no lucrativo, es que trabajan frecuentemente con escasez de recursos económicos y/o humanos. Además, las ONGs llevan a cabo actividades específicas, especialmente de coordinación y comunicación, que establecen necesidades concretas.

En las ONG, el uso del correo electrónico es frecuente, y supone un volumen considerable del trabajo que deben realizar personas voluntarias, cuyo tiempo es escaso, por lo que conseguir un aumento de la productividad en este aspecto es clave. A pesar de esto, apenas existen trabajos sobre la gestión del correo en el caso de las ONGs, si bien podemos destacar el desarrollado en [17], en el que se exponen pautas para mejorar la gestión del correo electrónico en el entorno de este tipo de organizaciones.

Con lo anterior en mente, en este trabajo nos centramos en la elaboración de una **prue-**

ba de concepto para mejorar el uso del correo electrónico dentro de una ONG. Para ello utilizaremos **técnicas de procesamiento de lenguaje natural** y **algoritmos de aprendizaje automático** para la clasificación de correos y la automatización o facilitación de tareas. Dicha prueba se propone para un caso real, el de la ONG *Labdoo*.

1.2. Objetivos y plan de trabajo

El propósito principal de este trabajo consiste en realizar una **prueba de concepto** que pueda ayudar a Labdoo a hacer un uso más eficiente del correo electrónico. Para ello, nos planteamos como **objetivo** identificar e implementar el conjunto mínimo de funciones que permita su desarrollo. Para conseguirlo, planteamos el siguiente plan de trabajo:

1. **Estudio de la gestión del correo en Labdoo:** En primer lugar, se analizarán las características y necesidades de la organización. En particular, se estudiará el uso que hace del email, para poder identificar los cuellos de botella y los principales problemas con los que se encuentran los/as voluntarios/as. Para ello, se trabajará con datos e información proporcionada por Labdoo [1].

2. **Examen preliminar:** A continuación se barajarán diferentes opciones, y se analizará su potencialidad en base a las necesidades prioritarias de la organización, así como sus recursos humanos y materiales, contando con el *feedback* del personal. El *output* de este análisis será la definición de unos casos de uso.

3. **Implementación:** Posteriormente se seleccionarán los casos de uso más relevantes y se llevará a cabo su implementación en forma de prueba de concepto. En particular, se desarrollará una API que permita manejar y extraer información útil de los correos,

[1] Aunque Labdoo colabora en el proyecto proporcionando los datos y facilitando información de su problemática, se trata de un proyecto totalmente ajeno a la organización.

clasificarlos, o modificarlos para que los/as usuarios/as de Labdoo puedan procesarlos en un tiempo menor al que les lleva en estos momentos.

4. **Análisis del funcionamiento:** Por último se estudiarán los resultados obtenidos para conocer la viabilidad de la propuesta, las funcionalidades disponibles y las vías futuras de trabajo.

1.3. Estructura de la memoria

En las secciones anteriores se ha definido el problema que se desea resolver, y se ha presentado su contexto y motivación. Además se han establecido los objetivos y el plan de trabajo. Con esto en mente, el resto de la memoria se organiza como sigue.

El Capítulo 3 introduce el **estado del arte** para los sistemas de gestión del flujo de trabajo. En primer lugar, se explican este tipo de sistemas, su uso y principales características. Posteriormente, se estudia la influencia que tiene el correo electrónico en ellos, se muestran los casos existentes más relevantes y se exponen las ventajas que presenta el uso del correo en la gestión del flujo de trabajo. Por último, se introducen los últimos avances de la Inteligencia Artificial aplicados a la clasificación de correos.

En el Capítulo 4 se presenta el **análisis** de la problemática y la solución propuesta. Para ello, en primer lugar se detalla el contexto específico y el funcionamiento de *Labdoo*. A continuación se define el problema a resolver y la solución seleccionada. Finalmente se detallan los casos de uso contemplados y el funcionamiento del sistema.

El Capítulo 5 introduce el tratamiento que se ha realizado sobre los **datos** proporcionados. Este preprocesamiento permite el posterior uso de los datos durante la fase de

desarrollo. También se presenta un análisis de la muestra final seleccionada. En base a este análisis, se establecen las líneas de trabajo que mejor se ajustan a los datos disponibles para la consecución de los objetivos.

El Capítulo 6 presenta el **desarrollo** realizado para la prueba de concepto. En primer lugar, se presentan los principales componentes del sistema y su diseño. Seguidamente se detalla la implementación de los casos de uso contemplados y las funciones desarrolladas. Finalmente, se muestra la **experimentación** llevada a cabo y los resultados obtenidos.

El Capítulo 7 recoge las principales **conclusiones** del trabajo. En él se resume el trabajo realizado, se analizan los resultados obtenidos y se proponen diversas mejoras a realizar y futuras líneas de trabajo.

Capítulo 2

Introduction

2.1. Presentation and motivation

Currently, despite the availability of social networks, workflow management systems, or Web 2.0, **email** still has an important weight in communication at all levels. In fact, its average use during the working day is estimated at 2.6 hours [12]. In many organizations, it is a habitat [14] where work is received, managed and delegated, and it is used for a variety of tasks, such as management of information, coordination and collaboration, or for contact management.

This extended use of email increases the amount of data that users have to process, causing high levels of stress and anxiety [13] [20] which makes them lose effectiveness when using it. The concept of *email overload* is defined as the feeling of being overwhelmed by the constant flow of messages that appear in the inbox, and the inability to handle their high volume efficiently [20]. This feeling is intensified by the continuous interruption of tasks, workflow and thinking [16] [18].

Given this problem, one of the most direct solutions for users is to employ different techniques to address messages more quickly, such as prioritizing, grouping or filtering emails.

In fact, new approaches have recently been defined aimed at increasing productivity, such as *inbox zero*, which proposes a series of guidelines to improve email management by users [7].

Advances in the fields of Artificial Intelligence and Information Retrieval are likely to help in mail management. These types of technologies have already been used, for example, for the prioritization of tasks or for the classification of emails [15] [21]. However, there are still no tools available that implement these techniques along with email client functionality, such as translation or filtering. In addition, the wide variety of tasks for which mail is used makes the development of general solutions a complex and non-trivial task, since the needs depend on the type of user and organization.

In this work, we focus on the automation of tasks and the improvement of email management in the specific case of **NGOs**. The main characteristic of this type of organizations, due to its non-profit nature, is that they frequently work with a shortage of economic and human resources. In addition, NGOs carry out specific activities, especially involving coordination and communication, that establish particular needs.

In NGOs, the use of email is frequent, and involves a considerable amount of work that must be done by volunteers, whose time is scarce, so getting an increase in productivity in this aspect is key. Despite this, there is hardly any available work on the management of the mail in the case of NGOs, although we can highlight the one developed in [17], which sets out guidelines to improve email management in the environment of this kind of organizations.

With the above in mind, in this work we focus on the development of a **proof of concept** to improve the use of email within an NGO. For this, we will use **natural language processing techniques** and **machine learning algorithms** for the classification of emails and the automation or facilitation of tasks. The proof is proposed for a real case, that of

the NGO Labdoo.

2.2. Objective and work plan

The main purpose of this work is to conduct a **proof of concept** that can help Labdoo make more efficient use of email. To do so, we aim to identify and implement the minimum set of functions that allow its development. In order to achieve this, we propose the following work plan:

1. **Study of mail management in Labdoo**: First, the characteristics and needs of the organization will be analyzed. In particular, the use of email will be studied, in order to identify the bottlenecks and the main problems that volunteers face. For this, we will work with data and information provided by Labdoo.

2. **Preliminary examination**: Different options will be considered, and its potential will be analyzed based on the priority needs of the organization, as well as its human and material resources, with the feedback of the staff. The output of this analysis will be the definition of several use cases.

3. **Implementation**: Subsequently, the most relevant use cases will be selected and their implementation will be carried out as a proof of concept. In particular, an API which allows handling and extracting useful information from emails, classifying them, or modifying them will be developed so that Labdoo users can process them in less time than what is currently taking them.

4. **Performance analysis**: Finally, the results obtained will be studied to determine the feasibility of the proposal, the available functionalities and future work paths.

2.3. Document structure

In the previous sections, the problem to be solved has been defined, and its context and motivation have been presented. In addition, the objectives and work plan have been established. With this in mind, the rest of the document is organized as follows.

Chapter 3 introduces the **state of the art** for workflow management systems. First, this type of systems, their use and their main characteristics are explained. Subsequently, the influence that email has on them is studied, the most relevant existing cases are shown and the advantages of the use of email in workflow management are presented. Finally, the latest advances in Artificial Intelligence applied to mail classification are introduced.

Chapter 4 presents the **analysis** of the problem and the proposed solution. In order to do this, the specific context and operation of *Labdoo* is detailed in the first place. Next, the problem to be solved and the solution selected are defined. Finally, the contemplated use cases and the operation of the system are detailed.

Chapter 5 introduces the preprocessing that has been performed on the **data** provided. This preprocessing allows the subsequent use of the data during the development phase. An analysis of the selected final sample is also presented. Based on this analysis, the lines of work that best fit the data available for achieving the objectives are established.

Chapter 6 addresses the **development** carried out for the proof of concept. First, the main components of the system and its design are presented. The implementation of the use cases contemplated and the functions developed are detailed below. Finally, the **experimentation** carried out and the obtained results are shown.

Chapter 7 gathers the main **conclusions** of the work. It summarizes the work done,

analyzes the obtained results and proposes various improvements to be made and future lines of work.

Capítulo 3

Estado del arte

En esta sección se estudia el concepto de *workflow management* (gestión del flujo de trabajo) y el papel que ejerce el correo electrónico en éste. Se muestra la importancia que ha adquirido en las empresas y organizaciones actuales y las ventajas que presenta el correo electrónico para que las ONGs puedan enfrentarse a este problema. Se revisan técnicas y casos de estudio exitosos en este ámbito que servirán para realizar la propuesta de la solución.

3.1. Gestión del flujo de trabajo

La gestión del flujo de trabajo consiste en proporcionar soporte automatizado para los procesos de negocios [22]. Un proceso de negocio es un conjunto de tareas o etapas a través de las cuales se encauza el trabajo. Estos flujos de trabajo están compuestos tanto por personas como por herramientas tecnológicas, de manera que la información correcta llega a la persona indicada en el tiempo adecuado.

La definición, ejecución, registro y control de procesos son el objetivo principal de un sistema de gestión de flujo de trabajo [25]. Debido al importante papel que desempeñan los procesos en la administración del flujo de trabajo, es importante utilizar un marco establecido para modelar y analizar el proceso del flujo de trabajo.

Las nuevas tecnologías han contribuido a la aparición de numerosos sistemas de este tipo [8] que se han convertido en herramientas fundamentales en las empresas actuales para mejorar sus procesos y ser competitivas. Sin embargo, estos requieren una alta inversión tecnológica [24] que en muchos casos las ONGs no pueden asumir dados sus limitados recursos. Es posible realizar una gestión del flujo de trabajo sin estos sistemas, aunque su uso tiene ventajas evidentes, como una mejor respuesta ante cambios en el entorno, la modificación de la legislación o la evolución de las condiciones del mercado [22].

3.2. Gestión de flujo de trabajo a través del correo electrónico

Uno de los principales significados de *management by email* es el uso del correo electrónico como herramienta principal en la comunicación de una organización [24]. Hoy día debido a las ventajas que tiene esta tecnología, como la facilidad de uso, su gran implantación o los reducidos costes económicos hace que la mayor parte de las organizaciones lo utilicen para tal fin.

Las grandes empresas pueden permitirse sistemas empresariales para una gestión exitosa de los procesos de negocio, pero no es el caso para las PYMES y ONGs. Los bajos costes económicos del correo electrónico han hecho que aparezcan nuevos sistemas basados en éste, como el trabajo realizado en [12], donde se propone un sistema de control de flujo que lidia con la gran flexibilidad y el carácter desestructurado del email. El sistema convierte el correo electrónico en un marco de gestión de procesos estructurado en el que los emails entrantes son asociados con procesos de negocio y se mejoran con anotaciones sensibles al contexto como los pasos siguientes a efectuar más recomendables.

En el modelo de *management by email* presentado en [24] los procesos de negocio son realizados únicamente a través de la comunicación vía email y las transiciones entre las etapas de estos procesos se inician al recibir o enviar un correo electrónico, de esta manera cualquier transición de un proceso de negocio, tiene asociado un email. En este modelo, el correo electrónico es suficiente para almacenar la información relevante de un proceso y es aconsejado para organizaciones en las que el correo electrónico ya se usa para la comunicación y en la que los procesos son conocidos y están definidos.

La propuesta presentada en [19] integra un sistema de seguimiento (*tracking system*) en el correo electrónico para rastrear las tareas hasta que éstas son completadas, y en caso contrario informar a las personas necesarias en los momentos oportunos. Cuando se envía un correo se asocia con un *issue*, para el que se crean URLs accesibles por el destinatario y a través de las cuales se realiza el seguimiento.

El papel fundamental que desempeña el correo electrónico en la gestión de tareas, se discute en los trabajos presentados en [9] [10] donde se sugiere que el principal elemento de interés son las tareas. Aquí se propone el término *thrask*, o conjunto de mensajes que hacen referencia a una tarea y que son agrupados analizando los mensajes de entrada. Además, se muestra que es posible afectar positivamente la experiencia del correo electrónico incorporando los recursos de administración de tareas directamente en la bandeja de entrada.

También se han realizado sistemas inteligentes para la gestión del flujo de trabajo basados principalmente en el texto del mensaje. En el trabajo realizado en [11] los usuarios pueden plantear consultas o tareas al sistema, el cual es capaz de desencadenar los subprocesos necesarios. De esta forma la solución se plantea como un diálogo con el sistema el cual utiliza el procesamiento del lenguaje natural como base para su funcionamiento.

3.3. Procesamiento de contenido de correo electrónico y clasificación

Dado que muchas tareas vienen dadas en forma de emails, los usuarios tienden a usar los clientes de correo electrónico como un administador de tareas y a verlo como una lista de cosas por hacer [15]. Sin embargo, los clientes no disponen de la funcionalidad necesaria para realizar estas tareas de forma eficiente. Los usuarios encuentran dificultades a la hora de hacer uso de las funcionalidades básicas de los clientes de correo electrónico y adaptarlas a su situación concreta. En el trabajo realizado en [15] se describe un sistema híbrido entre cliente de correo electrónico y administrador de tareas, que hace uso de la Inteligencia Artificial para, entre otras funcionalidades, hacer un cálculo para la priorización de mensajes con resultados exitosos.

Ya en el trabajo presentado en [24] se aconseja la extensión de la funcionalidad básica de los clientes de correo electrónico para conseguir un máximo rendimiento en la gestión del flujo de trabajo. En general, la principal herramienta para la gestión de correo electrónico es la clasificación automática de correo electrónico [21]. Un clasificador de correo electrónico es un sistema que clasifica automáticamente emails en unas categorías predefinidas. Por ejemplo, puede utilizarse para clasificar un email como spam o no, o para asignar etiquetas. La estructura del proceso de clasificación se divide en tres partes, tal y como se observa en la Figura 3.1: preprocesamiento, aprendizaje y clasificación.

Cada una de las tres partes principales en las que se divide la clasificación está compuesta a su vez por procesos más sencillos. El preprocesamiento comienza con la tokenización, a través de la cual se identifican las unidades básicas a ser procesadas, las palabras. A continuación, se eliminan las *stop words*, o palabras que más se repiten en un idioma (a, el, cuál, aquel...) y que aportan poco valor al proceso de aprendizaje. Esto mejora el rendimiento del clasificador en tiempo de procesamiento y en la tasa de acierto.

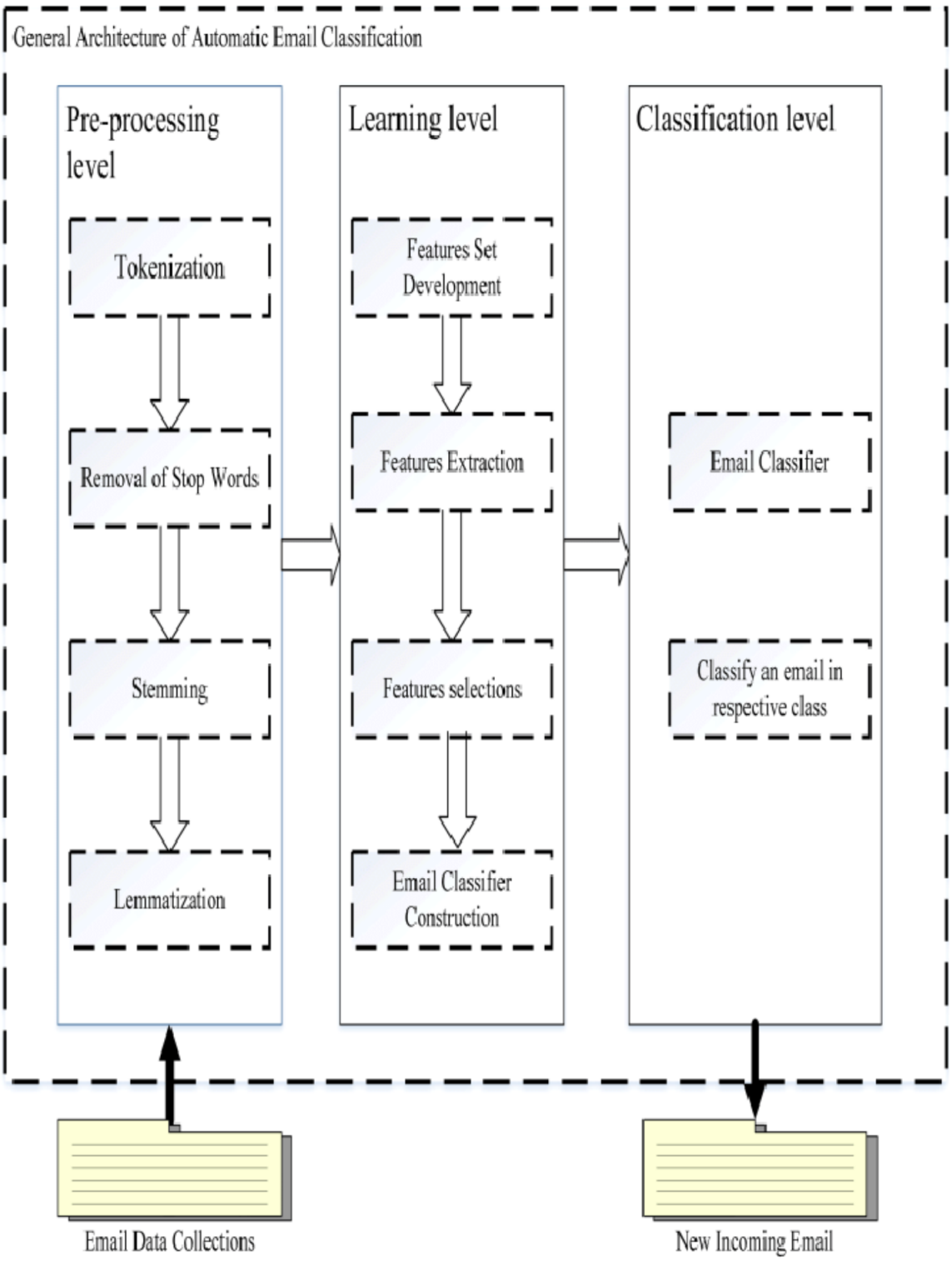


Figura 3.1: Arquitectura general de la clasificación automática de correos [21].

Stemming y *lemmatization* son técnicas usadas para normalizar las palabras. Las lenguas están formadas por múltiples palabras, muchas de ellas derivadas de otras (teléfono, telefonillo, telefonear...) que comparten una forma base (telefon). Normalizar las palabras consiste en encontrar esta forma base. Los algoritmos de *stemming* lo hacen cortando el final o el principio de la palabra, teniendo en cuenta una lista de prefijos y sufijos comunes para una forma base. Los algoritmos de *lemmatization* por otro lado, son más complejos y toman en consideración el análisis morfológico de las palabras.

Los algoritmos de aprendizaje automático requieren que todos y cada uno de los documentos textuales estén representados en la forma de un vector para comenzar a entrenar, esto se realiza a través de un modelo de espacio vectorial. Estos modelos se implementan a través del modelo *bolsa de palabras* usando TFIDF o mediante word embeddings. El modelo de bolsa de palabra es el más extendido por su facilidad de uso y su eficiencia computacional. En este modelo, cada documento parece una bolsa que contiene algunas palabras, permitiendo un modelado basado en diccionarios, donde cada bolsa contiene unas cuantas palabras del diccionario [5].

En el nivel de aprendizaje se extrae el conjunto de *features*, que son propiedades medibles o características de los correos. En la clasificación de correos, la extracción del conjunto de *features* influye determinantemente en el proceso de aprendizaje. De este conjunto se seleccionan las características más discriminativas para la clasificación con el objetivo de mejorar el rendimiento del clasificador en términos de precisión y eficiencia [21].

Las técnicas de aprendizaje supervisado son las más utilizadas en clasificación de correos electrónicos [24]. En estas técnicas, se proporciona a los algoritmos con un conjunto de instancias de entrada y unas clases de salida para identificar una función que aproxime

de manera generalizada este comportamiento. Los algoritmos de este tipo más comunes son: máquinas de soporte vectorial (*Support Vector Machines, SVMs*), árboles de decisión, redes neuronales, Naive Bayes, redes bayesianas y random forest. Las SVMs son los algoritmos que mejor se comportan y son las técnicas más utilizadas para la clasificación de emails [24]. Finalmente se entrena el clasificador y se guarda de manera que se puedan clasificar correos entrantes.

La clasificación de correos electrónicos basada en el contenido de estos utiliza palabras clave. Las técnicas de aprendizaje estadístico asignan una puntuación a estas *keywords*, y la puntuación total es utilizada para clasificar los emails entrantes.

La exactitud en las predicciones se ve afectada por la cantidad de instancias de entrenamiento, a mayor conjunto de datos de entrenamiento mayor la precisión que obtendrá el algoritmo. Etiquetar el conjunto de datos de entrenamiento es uno de los principales problemas de las técnicas de aprendizaje supervisado, y puede llegar a requerir una gran cantidad de tiempo.

Validación cruzada es la técnica más extendida para evaluar el rendimiento de un clasificador. Ésta consiste en dividir el conjunto de datos de entrenamiento en partes equivalentes (por ejemplo, 4 conjuntos con el 25 % de los datos cada uno). Por cada conjunto se entrena un clasificador con los conjuntos de datos restantes (75 % de los datos) y se procede a la clasificación del conjunto inicial con el modelo entrenado [6]. Finalmente se compara las predicciones del clasificador con las etiquetas asignadas a mano.

Con los resultados anteriores se definen métricas para evaluar el rendimiento del modelo, como se observa en la Figura 3.2:

- *Accuracy*: porcentaje de instancias para las que se ha predicho la etiqueta correcta.

- *Precision*: porcentaje de instancias que el clasificador obtuvo correctamente de la cantidad total de ejemplos que predijo para una etiqueta determinada.

- *Recall*: porcentaje de ejemplos que el clasificador predijo para una etiqueta dada del número total de ejemplos que debería haber predicho para esa etiqueta.

- *F1 score*: media harmónica de *precision* y *recall*.

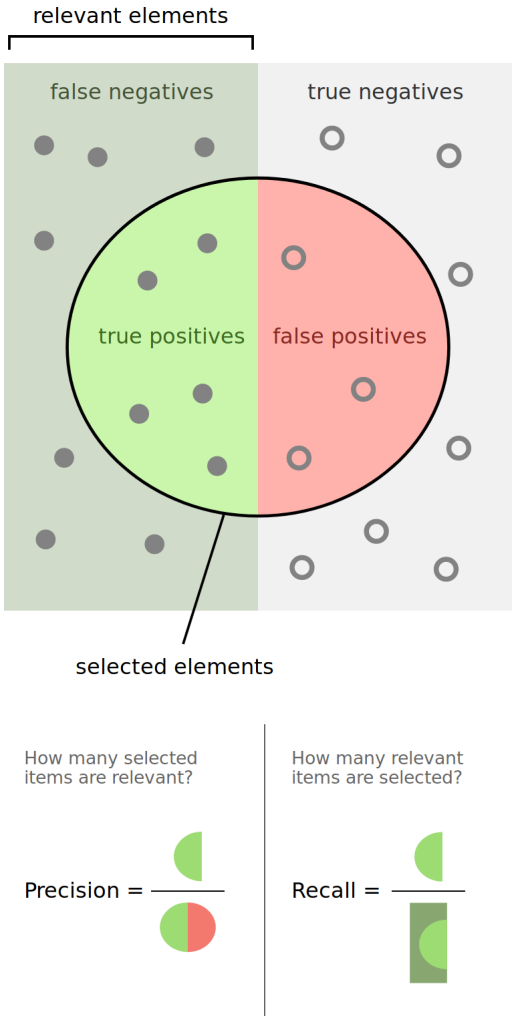


Figura 3.2: *Precision* y *recall* [3].

Capítulo 4

Análisis y propuesta

Labdoo recibe una gran cantidad de correos electrónicos para comunicarse con la gente que desea colaborar. Responder todos estos emails supone mucho trabajo a los voluntarios de Labdoo. Se pretende realizar la resolución al problema de *email overload* a través de un modelo de gestión basado en el correo electrónico. En concreto, se pretenden alcanzar los objetivos extendiendo la funcionalidad básica de los clientes habituales.

Utilizando este enfoque, el sistema se adapta a las restricciones de Labdoo en varios sentidos. El correo es económico y es la forma que ya se viene usando en la organización para comunicarse con la gente. No incurre en costes para Labdoo, como podrían llegar a suponer otras soluciones, como un asistente conversacional, el cual requeriría intervenir en los servidores y aumentar la carga de estos. Es fácil de utilizar, y las personas suelen estar familiarizadas con él. Además, se plantea como una de las opciones más sencillas de implementar.

En general se trata de proporcionar un conjunto de funcionalidades que libere a los voluntarios de Labdoo de una parte de su trabajo y que aumente la productividad en aquél que tengan que hacer. La liberación de trabajo se consigue mediante la respuesta automática de emails y el aumento de la productividad proporcionando métodos no disponibles en los clientes de correo comunes.

4.1. Contexto

Labdoo es una red social humanitaria formada por personas de todo el mundo con el objetivo de llevar una mejor educación a regiones donde donde se necesite. La forma de colaborar con la educación de estas regiones es dotarlas de dispositivos electrónicos cargados de software educativo. Esto se consigue a través de voluntarios que pueden participar en distintas tareas, tales como donar dispositivos, sanearlos, transportarlos, etc. Como se indica en [4], Labdoo:

- Es sin fines de lucro.
- Es colaborativa.
- Está totalmente distribuida a lo largo del mundo.
- El código que ejecuta la red social es abierto y gratuito. (El código está disponible en `https://github.com/labdoo`).
- Cualquiera puede participar.
- Está diseñado sin incurrir en emisiones de CO_2 adicionales.
- No requiere financiación para funcionar.
- Está libre de cualquier tipo de publicidad.

En la figura 4.1 se muestran los componentes de Labdoo. Los elementos principales en los que se basa la actividad de la organización son los *dootronics*. Puede considerarse como tal cualquier equipo electrónico (portátil, smartphone, tablet...) con unas especificaciones técnicas mínimas a los que se les puede instalar software educativo.

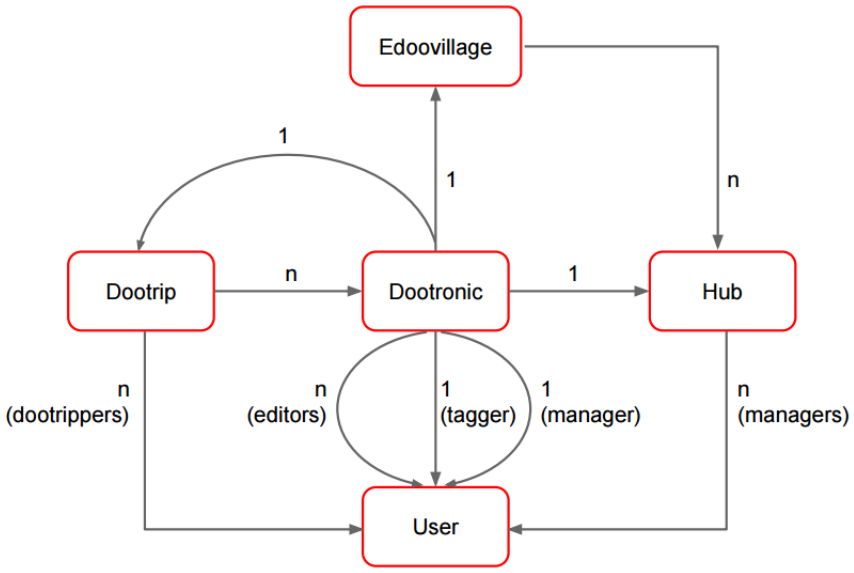


Figura 4.1: Elementos de Labdoo y su relación [4].

A los usuarios de la red social se les conoce como *labdooers* y a los centros educativos que reciben los dootronics para su uso final como *edoovillages*. Es importante destacar que los dootronics no son asignados a particulares, sino a escuelas. Los dootronics son enviados a su destino final a través de *dootrips*, que son realizados por voluntarios, turistas, estudiantes internacionales, etc. Para mejorar la organizacion de los labdooers se utilizan los *hubs*, formados por grupos de voluntarios de una misma región.

En la Figura 4.2 se muestra el ciclo de vida de los dootronics. La vida de estos empieza cuando alguien decide donarlos etiquetándolo con los pasos que marca la página web de la organización. Tras esto, el dispositivo se sanea de datos innecesarios y se carga con el software educativo correspondiente, que suele realizarse en los hubs. Una vez el dispositivo está listo se busca un dootrip para transportarlo a la escuela que lo necesite. En este paso, los hubs mencionados anteriormente ayudan a que llegue a un destino buscando personas que quieran llevar el dootronic allá donde se haya solicitado, y también se pide colaboración a las escuelas para buscar dootrippers. Finalmente, los dispositivos llegan a los edoovillages, que han de estar registrados en la página web.

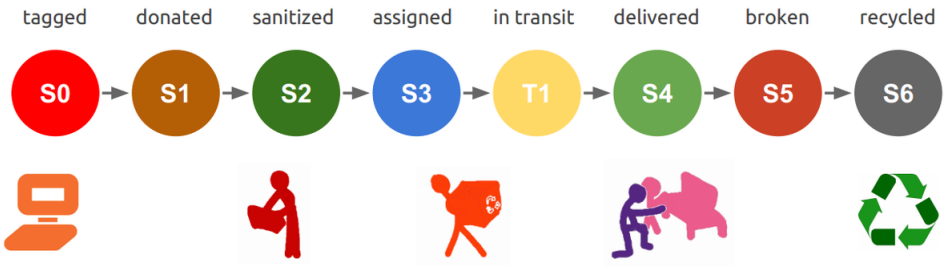


Figura 4.2: Ciclo de vida de un dootronic [1].

La labor social tiene varios objetivos: ayudar al desarrollo y educación de áreas desfavorecidas, reducir la brecha digital, reutilizar dispositivos electrónicos que en países desarrollados se dejan de utilizar y reciclarlos una vez su vida útil ha finalizado.

4.2. Análisis de la problemática

La principal forma de contactar con Labdoo es a través de un formulario de contacto (Figura 4.3). Una vez se ha establecido la comunicación, ésta continúa como un hilo de correo electrónico. El formulario de contacto tiene como campos obligatorios, además del cuerpo del mensaje, el país y la razón por la que se contacta a Labdoo, para la cual se permite indicar varias opciones:

 I want to contribute a laptop or a tablet (dootronic)

 I want to contribute a trip (dootrip)

 I want to become a Labdoo volunteer (labdooer)

 I want to solicit laptops or tablets for my educational project (edoovillage)

 I just want to send a message to Labdoo.org

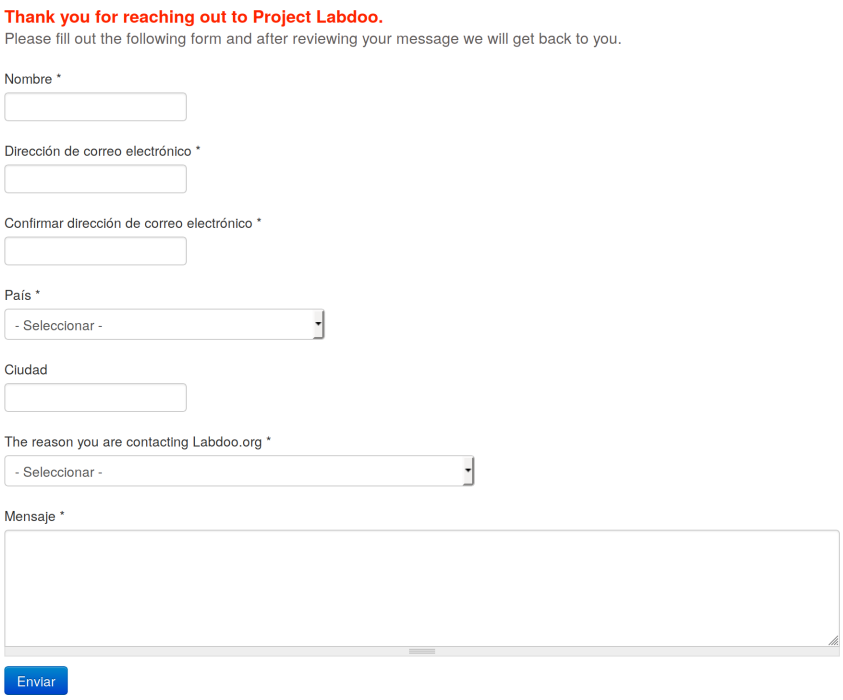


Figura 4.3: Formulario de contacto de Labdoo.

El único recurso del que disponen los voluntarios en estos momentos son plantillas, de manera que cuando reciben una solicitud, clasifican el problema que los usuarios refieren en una de ellas (si es que hay alguna acorde) y modifican algunos datos, como el país y el nombre del solicitante. Dado que la mayoría de los emails que reciben son para solicitar equipos electrónicos, una tarea importante es decidir si el uso que se le va a dar a los equipos informáticos es compatible con el objetivo de Labdoo. Las plantillas con las que cuentan los voluntarios son:

- Response to Dootronics Solicitation
- Response to Dootronics Donors
- Response to a Dootrip Offering
- Response to tax receipt request
- Response When You Cannot/Did Not Meet Delivery Deadline

- Response to an Unreasonable Request
- Response when Applicant does not fulfil Labdoo Requirements
- Response to request for information
- Forwarding a Response

Los problemas que se plantean a la hora de contactar con Labdoo son muy diversos. No obstante, una gran parte responden a consultas muy parecidas, y son unos pocos los que plantean cuestiones variadas. Los principales problemas con los que se encuentran voluntarios de Labdoo son:

1. Muchas personas no encuentran la información que necesitan o no son capaces de intuir cómo funciona el proceso a partir de la información que se facilita en la Web y hacen preguntas básicas, cuya respuesta normalmente podrían encontrar localizando la URL adecuada. Esto deriva en un gran número de correos similares que podrían responderse indicando los siguientes pasos que debe realizar el usuario y cómo ejecutarlos.

2. Algunas personas, solicitan ordenadores para darle un uso no acorde a los objetivos de Labdoo, como utilizarlos con fines no educativos u obtener un beneficio económico con ellos. Detectar estos casos es uno de los principales contratiempos.

3. El proceso de solicitar dootronics implica enviar datos a Labdoo a través del correo electrónico. Muchos de los usuarios no aportan toda la información requerida y esto supone un sobresfuerzo a los voluntarios, que tienen que volver a solicitar los datos.

4. El tiempo de los voluntarios es reducido y muy valioso puesto que su motivación es una labor social y no un rédito económico. Optimizar el tiempo de los voluntarios de manera que se maximize su contribución y se reduzca el esfuerzo que tienen que realizar es básico.

Para lograr poner solución a estos problemas, es necesario tener en cuenta las circustancias de Labdoo. Se debe garantizar en la medida de lo posible que aquellos correos más importantes o que aportan mayor beneficio a la labor social sean respondidos primero. De esta manera se evita, por ejemplo, que potenciales donadores terminen por no entregar los dootronics a Labdoo.

Los correos que reciben están en múltiples idiomas. Los voluntarios pueden no conocer todos aquellos en los que se trabaja, por lo que la asistencia lingüística es una ayuda que podría facilitarles el trabajo. De esta manera, una solución efectiva implicaría unificar herramientas que los voluntarios pueden estar utilizando en estos momentos en una sola, como el cliente de correo electrónico, un traductor, o el navegador para copiar las plantillas.

La solución ideal implicaría liberar completamente a los voluntarios de la tarea de responder correos, que se pudieran resolver las dudas al instante y que las decisiones se tomen según los principios de Labdoo. Dado que esto es complicado de conseguir, cualquier solución que aumente la productividad de los voluntarios, de manera que tengan que dedicar menos tiempo a esta tarea, y reduzca el tiempo de demora de la respuesta es una gran mejora con respecto a la situación actual.

Además de para comunicarse con la gente que desea colaborar, los voluntarios de Labdoo también utilizan el correo para la comunicación interna. Sin embargo, el principal problema expuesto por los voluntarios se centra en el gran número de solicitudes externas que reciben, por lo que no es prioritario atender a los correos relativos a la comunicación interna, que no serán tenidos en cuenta.

4.3. Casos de uso

Generar estadísticas generales. Desde el inicio del sistema, el usuario puede generar información acerca del estado de su bandeja de entrada. Esta información ayuda a analizar cómo abordar los mensajes, estimar un tiempo de trabajo, establecer distintas estrategias y determinar qué filtros usar o los criterios de priorización a utilizar más adecuados. Así, se desglosa el número de correos por leer según el asunto, el número de correos para los que se detecta que se podrían responder automáticamente, el número de nuevas solicitudes de contacto (a través del formulario), y aquéllos que forman parte de un hilo o *issue* ya abierto, los idiomas utilizados, o los mensajes provenientes de personas pertenecientes a Labdoo.

Filtrar bandeja de entrada. El usuario puede filtrar los mensajes de su bandeja de entrada en base a varios criterios. Este mecanismo mejora la organización del usuario, le permite responder correos similares de forma consecutiva, responder primero aquéllos que considere más urgentes, etc. De esta manera es posible abordar el problema número 1 filtrando correos con cuestiones más básicas y responderlos posteriormente. También se afronta el problema 2 pudiendo cribar aquellos emails que solicitan ordenadores para hacer un uso no adecuado de estos. Distintos filtros pueden estalecerse, entre ellos:

- Mensajes recibidos a través del formulario.
- Asunto (dootronic, dootrip, labdooer, edoovillage, other).
- Acción recomendada.
- Dirección de correo electrónico de los remitentes.
- Dominio de las cuentas de correo de los remitentes.
- Nivel de prioridad.
- Palabras clave.

- Archivos adjuntos.

Priorización de correos. El usuario puede establecer distintos criterios para priorizar la bandeja de entrada. La prioridad puede establecerse en función del asunto (dootronic, labdooer, edoovillage...), el tiempo que el correo lleva sin leer, el número de correos enviados por el usuario, longitud del mensaje, la aparición de palabras clave, si procede de alguien en el libro de direcciones o una persona interna a Labdoo, la acción recomendada que se haya detectado, etc. Establecer criterios de prioridad adecuados, ayuda al usuario indirectamente con los problemas 1 y 2, permitiéndole responder aquellos correos que requieran de un menor tiempo de demora en la contestación, y en caso de no tener tiempo para reponder a todos, asegurar que estos queden respondidos.

Respuesta automática de correos. Desde el inicio del sistema, éste busca la acción recomedada para los correos electrónicos que aún no han sido leídos y que se escribieron desde el formulario de contacto. Estas acciones son aquéllas especificadas por las plantillas con las que los voluntarios de Labdoo cuentan para responder los correos electrónicos. Los emails se clasifican en una de estas plantillas o en ninguna en el caso de que no se ajusten al problema propuesto por el remitente. El usuario tiene dos opciones para utilizar esta acción recomendada. La primera es que después de leer el correo y comprobar que la recomendación es correcta, ejecutarla. De esta forma la respuesta de correos se realiza de forma semisupervisada. La segunda opción es indicar que se responda de forma automática todos aquellos emails para los que se hayan encontrado acciones recomendadas. Esta última opción ahorra tiempo a los voluntarios de Labdoo, pero tiene la desventaja de no validar las respuestas, pudiendo incurrir en errores. En mayor o menor medida, esta funcionalidad contribuye a que el usuario pueda mitigar los problemas definidos en la sección anterior.

Identificar formularios de contacto incompletos. Este caso de uso aporta una solución al problema 3. Las personas que desean solicitar dootronics por primera vez han de

completar un formulario que debe ser rellenado y enviado por correo (en el mismo cuerpo del email) a los voluntarios de Labdoo. Hay veces en las que faltan campos por completar, y los voluntarios deben comprobar uno a uno que todos ellos estén rellenos. Puesto que esto toma mucho tiempo a los usuarios, se pueden detectar aquellos correos que cuentan con un formulario, y de ellos, aquéllos que están incompletos. La acción recomendada en estos casos es un mensaje pidiendo que vuelvan a enviar el formulario con todos los campos obligatorios completados.

Ver histórico de una cuenta de correo. Los voluntarios de Labdoo al procesar un correo pueden necesitar recordar mensajes anteriores del remitente para poder tomar decisiones. Por tanto, el usuario tiene la posibilidad de recuperar el histórico de mensajes de una cuenta de correo electrónico. Los mensajen se devuelven al usuario ordenados de forma cronológicamente inversa. Éste y los siguientes casos de uso ponen solución al problema 4 facilitando a los usuarios sus tareas y optimizando su tiempo.

Ver información de una cuenta de correo. Similar a la funcionalidad anterior. Esta muestra información de una cuenta de correo electrónico como números de correos recibidos, sin leer, número de veces que ha usado el formulario, asuntos más frecuentes, país de origen, etc.

Traducción de correos. Hay varias lenguas que los remitentes utilizan para escribir los correos. Puesto que un voluntario de Labdoo no puede conocer todas ellas, tiene la opción de traducir el correo al inglés. Esto le permite analizar el mensaje y saber a quien redirigir el correo dentro de Labdoo. En caso de que sea el mismo voluntario el que quiera responder el email, éste puede escribirlo en inglés y posteriormente se traduce el cuerpo del mensaje a la lengua del destinatario.

Uso de firma. Los usuarios de Labdoo utilizan habitualmente una firma. Con el objetivo de ahorrar tiempo, pueden configurar una por defecto, de forma que se incluya automáticamente en el cuerpo de los emails.

Libro de direcciones. El usuario puede mantener un libro de contactos, que puede utilizarse posteriormente para filtrar la bandeja de entrada o priorizar. Así los voluntarios responden antes aquellos correos procedentes de personas afines y que requieran mayor urgencia en la respuesta.

Priorización de los casos de uso

A continuación se presenta una lista de los casos de uso ordenada de más a menos prioritarios. Para realizarla se ha tenido en cuenta tanto el beneficio que el caso de uso aporta al problema, como criterios de implementación:

1. **Respuesta automática de correos**.
2. **Generar estadísticas generales**.
3. **Ver histórico de una cuenta de correo**.
4. **Priorización de correos**.
5. **Identificar formularios de contacto incompletos**.
6. **Filtrar bandeja de entrada**.
7. **Ver información de una cuenta de correo**.
8. **Traducción de correos**.
9. **Uso de firma**.

10. **Libro de direcciones**.

La respuesta automática de correos es el caso más completo, pues requiere interactuar con los servidores IMAP y SMTP además del procesado de los textos de los emails y un modelo de clasificación de correos. Este caso es también el que libera de mayor carga de trabajo a los voluntarios. La generación de estadísticas generales e histórico de una cuenta comparten gran parte de la implementación y resuelven tareas que se presentan de manera recurrente.

La priorización de correos y la identificación de formularios de contacto incompletos son dos de los casos de uso que más facilitan el trabajo a los voluntarios. Se ha decidido reducir su prioridad ya que su implementación se ve obstruída al necesitar de Labdoo para definir un criterio de priorización, y la ausencia de correos con formularios en los datos con los que se cuentan.

Los casos de uso anteriores se entienden como los básicos para el sistema. La prueba de concepto a desarrollar contempla los tres primeros, pues son los más relevantes, y gran parte de su implementación es compartida con el resto de los casos.

4.4. Funcionamiento

El flujo que se hace de los mensajes teniendo en cuenta todos los casos de uso se observa en la Figura 4.4. En primer lugar, se extraen los parámetros del formulario en el caso de que el correo se escribiera a través de éste. En el caso contrario los parámetros se dejan en blanco. En segundo lugar, se traducen los mensajes si la traducción está activada. Para ello, se reconoce el lenguaje en el que está escrito el cuerpo y después se procede con la traducción si el idioma original difiere del inglés.

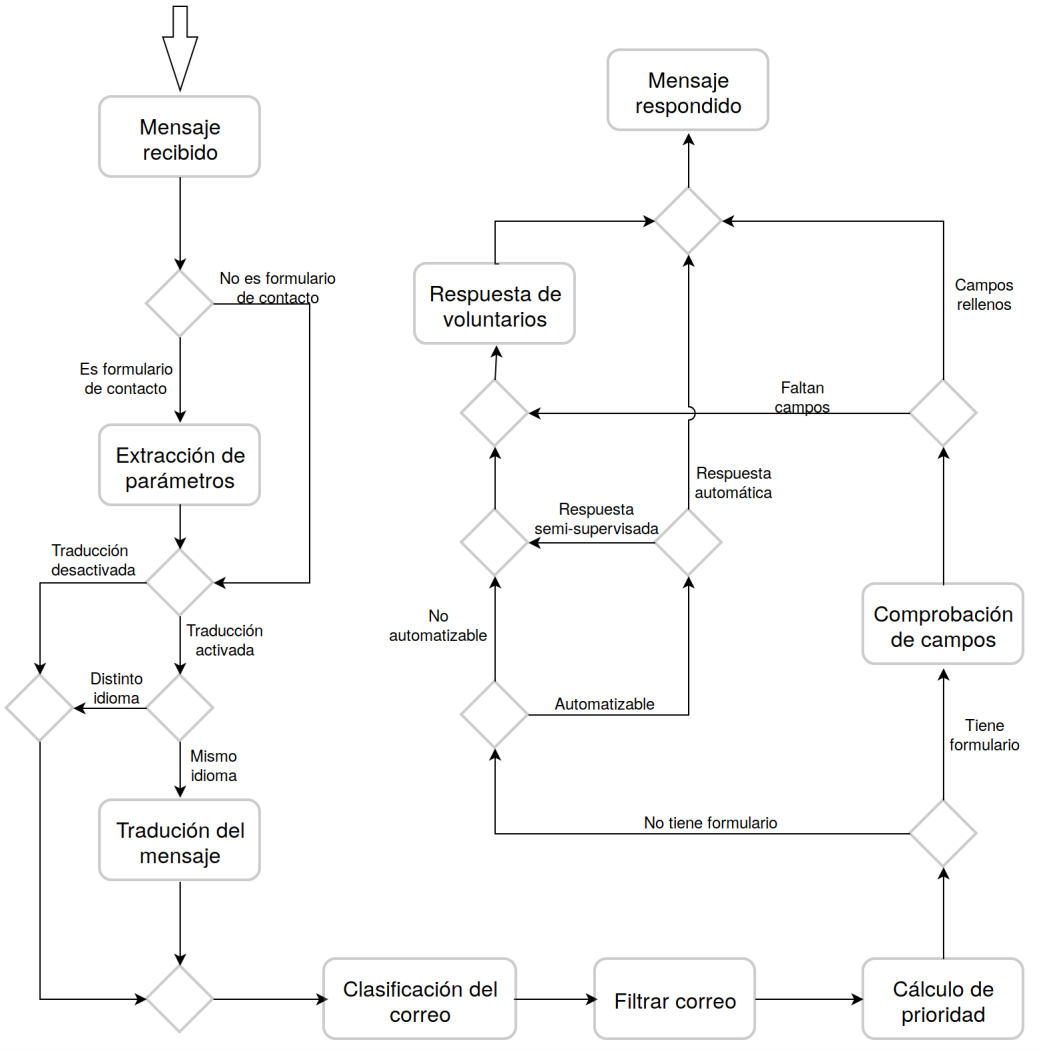


Figura 4.4: Procesamiento que se realiza de los correos.

A continuación se clasifican los mensajes. Para conseguirlo primero se parsea el asunto y se clasifica en una de los opciones indicadas en la Sección 4.2 u *otra* si no se corresponde con ninguna de ellas. Aquellos correos que coincidan con *edoovillage* son preprocesados y proporcionados al modelo de clasificación, que puede asignar dos etiquetas distintas: *solicitud* o *no_cumple_requisitos*.

Seguidamente, se realiza el procesado necesario para los casos de uso. Primero se filtra

el buzón de acuerdo a las características seleccionadas por el usuario. Después se realiza un cálculo de la prioridad de manera que aquéllos más urgentes quedan en lo alto del buzón.

Finalmente se procede a responder los correos automáticamente o a encolarlos de manera que el usuario los pueda responder uno a uno de forma semi-supervisada. Para que un correo sea respondido la configuración correspondiente debe estar habilitada. Hay dos vías por las que se detecta un correo a responder, o bien contiene un formulario de solicitud de *edoovillage* que tiene todos sus campos completos, o bien se corresponde a las categorías *dootronic* o *edoovillage*.

Capítulo 5

Tratamiento de datos

Labdoo ha facilitado una muestra de datos con 1554 correos en formato *mbox*, que constituyen los datos reales sobre los que se basará la propuesta. Como paso previo, se ha llevado a cabo un tratamiento de los mismos que permita trabajar con sus características y contenido.

5.1. Preprocesamiento de los datos

En la Figura 5.1 se muestra el esquema general del preprocesamiento de los datos. En primer lugar, todos los correos han sido anonimizados usando el script en [23]. Con este procedimiento se minimiza en gran medida la información personal presente en los emails. Si bien el script puede llevar a la pérdida de cierta información, consideramos que es necesario para adecuarnos a los objetivos éticos necesarios en la propuesta.

A continuación, para facilitar el tratamiento de la información se ha convertido el formato de los datos a *csv*. Éste presenta un uso más extendido y permite, entre otras opciones, trabajar fácilmente haciendo uso de librerías como *Pandas*.

El conjunto inicial de los datos presenta diversas incongruencias. En la medida de lo posible, se han tratado de solucionar mediante scripts, aunque en diversos casos ha sido necesario

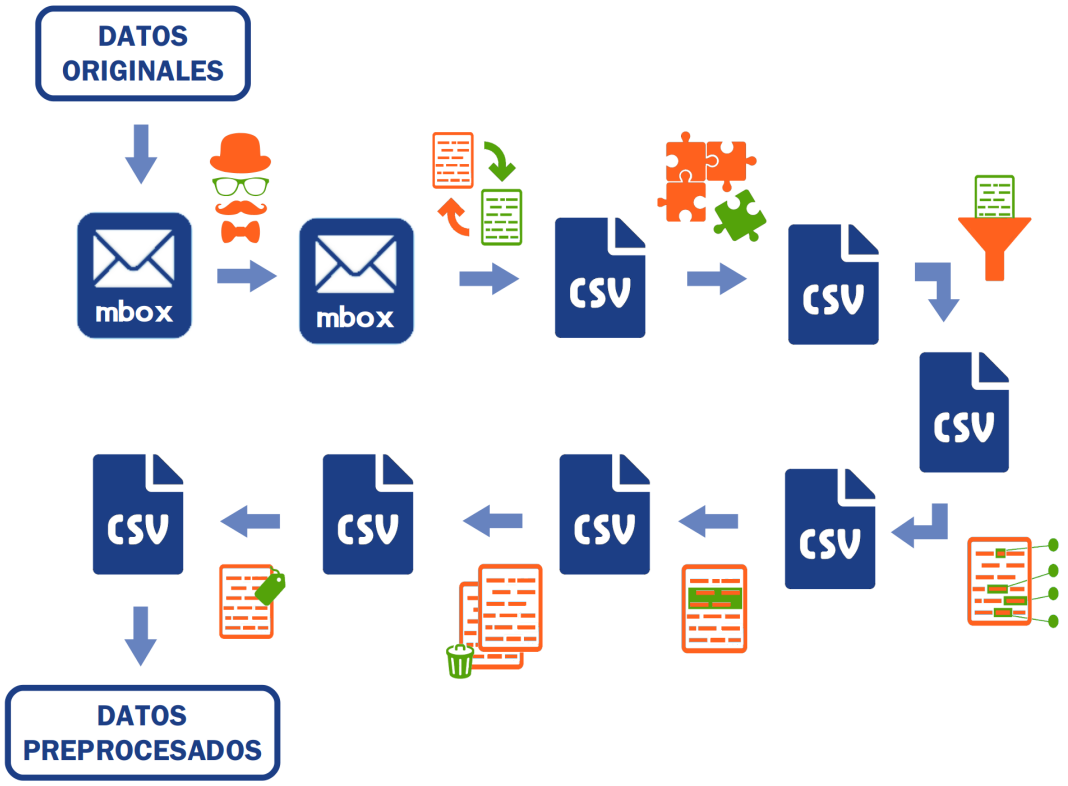


Figura 5.1: Esquema del preprocesamiento de los datos.

resolverlas a mano total o parcialmente. A continuación se presentan las incongruencias más frecuentes:

- Distintos mensajes utilizan nombres de campos distintos para hacer referencia al mismo. Por ejemplo, *Message-ID* y *Message-Id* representan el identificador del mensaje.
- Al hacer el dump de la base de datos o durante el proceso de anonimización, algunos caracteres (según los casos observados, aquellos que no tienen representación en *ASCII*) han pasado a estar codificados en *utf-8* hexadecimal. Como ejemplo, «é» está codificado como «=C3=A9». Se han identificado los casos que cumplen este patrón y se han convertido a su correspondiente carácter en *unicode*.
- También se han incluido «= » y «=\n» aleatoriamente dividiendo palabras. Casos como «ma= ñana» o «via= je» eran frecuentes.

- En el asunto y el cuerpo de muchos correos de respuesta se incluyen el carácter «>», en ocasiones apareciendo varios consecutivos. También se incluyen otras codificaciones de la respuesta de correos, como «—– forwarded message —–» en el cuerpo de los correos.

- En el cuerpo de los mensajes se incluyen secciones de texto codificado o identificadores que no resulta relevante para nuestro estudio, aumentan el volumen del dataset y dificultan trabajar con él.

- Aparecen distintos nombres para un mismo país. De esta manera, Grecia puede aparecer como «grecia» o « Grecia».

- El proceso de anonimización sustituye algunos datos que dejan de tener sentido para nuestro estudio, y deberían ser eliminados.

En el siguiente paso se han reconstruido los hilos de mensajes, asignando un identificador a cada hilo, y otro al orden de los mensajes dentro de cada hilo. Como se ha decidido trabajar sólo con los mensajes escritos a través del formulario, se ha filtrado el conjunto de los mensajes desechando aquellos que estando en un hilo, no son los primeros de éste. Después se ha procedido a filtrar a mano. Esto se ha hecho a través de la terminal, mediante un script que muestra el mensaje por pantalla para que el usuario indique si es válido o no. En este paso, un mensaje se considera válido si en alguna parte del cuerpo hay una estructura que indica que existe un contacto a través del formulario. Esto resulta fácil de identificar puesto que, en este caso, existe la siguiente codificación tipo:

- User's name: Ejemplo De Nombre

- User's email address: [.de](3D"mailto:.de")

- User's country: Spain

- User's city: Huelva
- Reason for contacting you: I just want to send a message to Labdoo.org
- Message: mensaje escrito por el usuario

Una vez validados los emails, se ha realizado un parseo de los mensajes para obtener los campos anteriores normalizados. Al igual que en pasos anteriores primeros se han extraído todos los posibles automáticamente, y posteriormente se han extraído los restantes a mano.

El formulario de contacto de Labdoo facilita varias opciones que poner como asunto, pero debido a los problemas antes mencionados, y a modificaciones que se van haciendo de estos (como la inclusión de «Re:» al responder un mensaje), la mayoría han acabado por diferir del asunto original. Se ha asignado a cada correo el valor inicial del asunto en el caso de que se haya podido parsear correctamente. Dificultades en el parseo ocurren por estar escritos en otros idiomas, la inclusión de caracteres como «>» al comienzo de cada línea en los mensajes de respuesta, etc. Aquellos mensajes de los que no se han podido extraer los campos correctamente han sido asignados un valor por defecto y posteriormente han sido parseados a mano.

Los mensajes incluyen muchos metadatos que es preferible eliminar del cuerpo del mensaje, como los propios de mbox. Puesto que a nosotros sólo nos interesa lo que el usuario escribió en sí (aquello que aparece inmediatamente después de «Message:»), se han parseado estos mensajes y se han validado manualmente. Con este propósito, en una terminal se van mostrando los mensajes ya parseados. Si el parseo es correcto se indica. En caso contrario, se realiza manualmente en un paso posterior.

Existen correos pertenecientes a un hilo con el campo «In-Reply-To» vacío, por lo que al extraer el mensaje original obtenemos mensajes repetidos. La eliminación de estos du-

plicados se ha realizado agrupando los mensajes por «user_name», de manera que sólo es necesario comparar si hay repeticiones mirando un número reducido de mensajes.

Finalmente, a aquellos correos que tienen por asunto *dootronic*, *labdooer* o *dootrip* se les asigna la plantilla correspondiente. Aquéllos que tienen por asunto *edoovillage* se clasifican en función del contenido del mensaje en una de las dos plantillas disponibles: solicitud, y no cumple los requisitos. En el caso de que no se ajusten a alguna de estas dos, se deja sin asignar.

Este intensivo proceso de tratamiento reduce la muestra considerablemente, en torno al 20 % del tamaño original. Los datos finales resultantes suman un total de 315 correos, con fechas entre 2017-2019.

5.2. Análisis de la muestra

Los datos tratados permiten dibujar un marco genérico de las principales necesidades de la organización. Esto permite establecer algunos ejes de priorización para la propuesta.

Los asuntos de los correos (Figura 5.2) muestran que existe un tráfico importante correspondiente a las instituciones que solicitan la ayuda de Labdoo. El siguiente asunto más frecuente corresponde a los donantes de material. Estas dos categorías suponen el 84 % del intercambio de emails. En contraste, las comunicaciones con potenciales labdooers y voluntarios para el traslado de material no llegan conjuntamente al 10 % del tráfico total.

Estos datos muestran claramente que la sobrecarga en los emails procede sobre todo de la organización de la solicitud y donación de dootronics, por lo que es éste aspecto el que debe ser automatizado de manera prioritaria. Los contactos de personal voluntario para

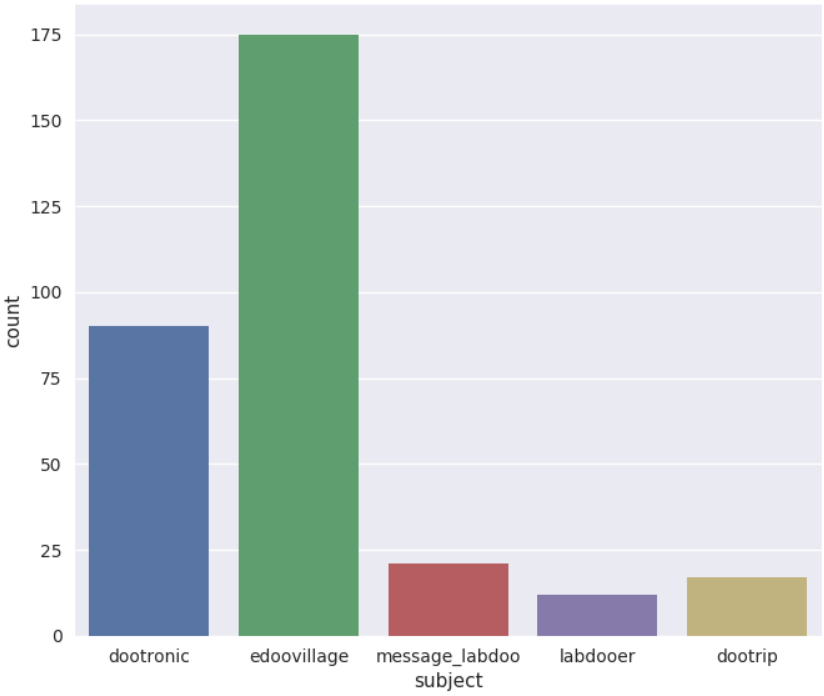


Figura 5.2: Frecuencias de los asuntos de los correos.

tareas de organización y traslado suponen un volumen mucho menor de trabajo. Además los asuntos que se tratan suelen requerir de una atención más personalizada. Por estos motivos, la automatización de estos emails es menos crítica para el objetivo planteado.

Otra de las dificultades que presenta la prueba de concepto es la variedad de lenguajes que se utilizan en los correos, que deben tenerse en cuenta en la solución propuesta. La Figura 5.3 muestra que la mayoría de emails, alrededor de un 70 %, se escriben en inglés.

Ante la variedad de idiomas en la que están escritos los correos, se ha decidido mantener el mensaje en su idioma original y realizar una copia en el idioma más extendido en la base de datos, realizando la traducción a través de *TextBlob*, que se basa en la API de traducción de Google [2].

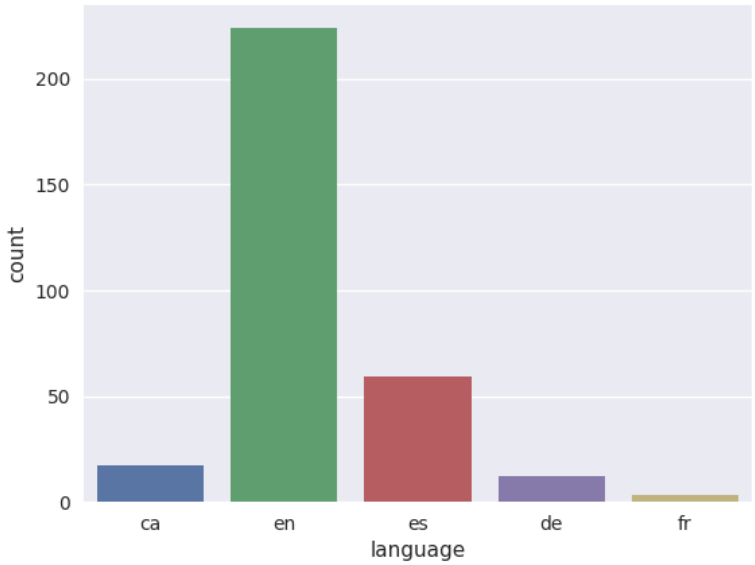


Figura 5.3: Frecuencias de los lenguajes utilizados.

Con respecto a la clasificación de correos, existen dos posibilidades: trabajar con los emails en su idioma original o utilizar la copia en inglés. En el primer caso, se evitan errores de traducción que puedan modificar sustancialmente el contenido. La segunda opción presenta la ventaja de un trabajo más homogéneo, que permite, por ejemplo, utilizar listados disponibles de stopwords en inglés. Además, puesto que no tendrían que hacerse clasificadores independientes por cada lenguaje, se proporciona al algoritmo un mayor número de instancias para el proceso de aprendizaje.

Como se oberva en la Figura 5.4 el etiquetado de los mensajes correspondientes a *edoo-village* resulta en un problema desbalanceado. La mayor parte de los mensajes cumplen con los requisitos de la organización frente a unos pocos que no pueden ser admitidos. Esto plantea problemas de cara al entrenamiento de un modelo que clasifique los emails en una de las dos categorías, siendo necesario recurrir a técnicas de *undersampling* u *oversampling*. Por otra parte, no ha sido posible clasificar 15 de los correos con este asunto, los cuales no

serán tenidos en cuenta.

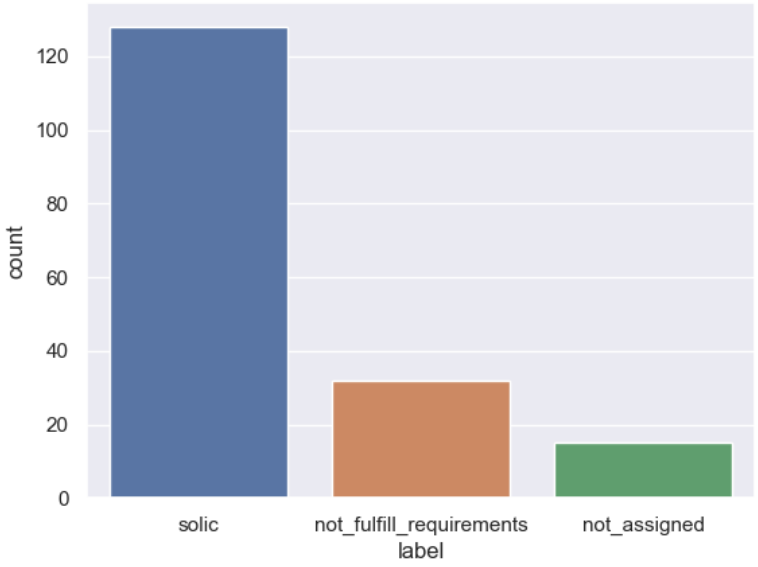


Figura 5.4: Distribución de las etiquetas asignadas a emails con asunto *edoovillage*.

Capítulo 6

Desarrollo

La funcionalidad a desarrollar se describe en la Sección 4.3. Los casos de uso considerados en la prueba de concepto son: generación de estadísticas generales, histórico de una cuenta y respuesta automática de correos. Este último es el que más interés presenta dado que es el puede aportar una mayor contribución a reducir la carga de trabajo de los voluntarios de Labdoo.

Cada caso de uso se resuelve mediante llamadas a funciones de APIs desarrolladas en *Python*. Estas APIs gestionan los distintos componentes, protocolos y procesamientos a ser abordados. Para comprender su funcionamiento, este capítulo explica el diseño, implementación y comportamiento del sistema.

6.1. Diseño

El esquema compartido de la interacción con el servidor IMAP que se realiza en los diferentes casos de uso se muestra en la Figura 6.1. El sistema comienza conéctandose al servidor a través de un socket cifrado e identifica al cliente mediante una contraseña en texto plano.

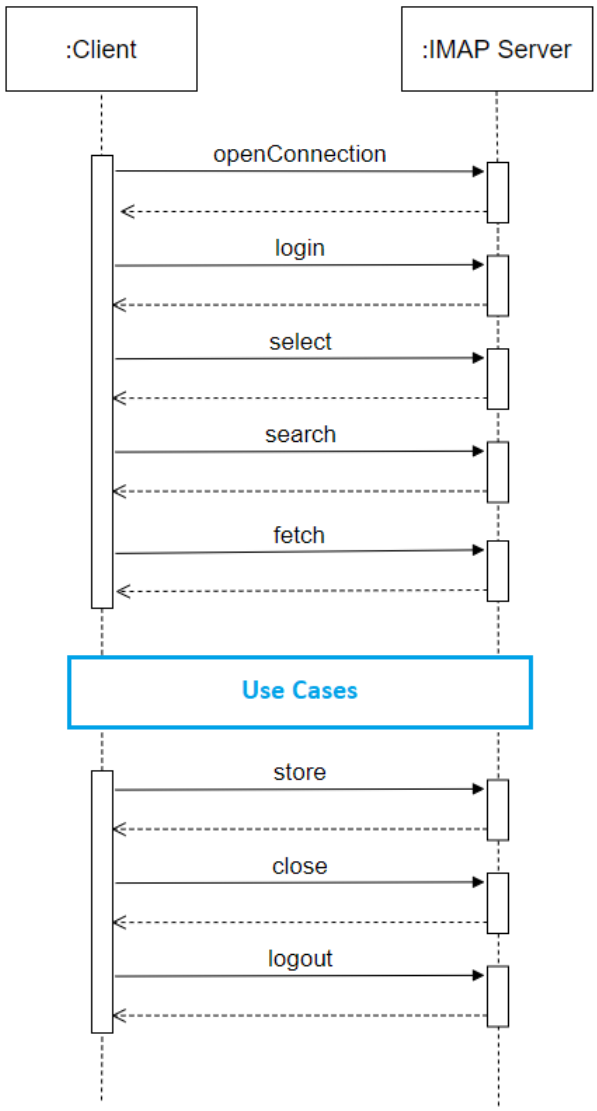


Figura 6.1: Esquema compartido de la interacción con el servidor IMAP.

La descarga de los correos del servidor se realiza a través de tres solicitudes: *select*, *search* y *fetch*. La primera selecciona el buzón del que se desean traer los mensajes. *Search* devuelve los identificadores de los mensajes coincidentes del buzón seleccionado de acuerdo a un criterio de búsqueda. Finalmente *fetch* realiza una petición al servidor proporcionando los identificadores devueltos por *search* y devolviendo los correos en crudo. Estas llamadas se realizan para el buzón de entrada, y *select* y *search* se realizan dos veces: una para todos los mensajes y otras para los mensajes no vistos. De esta forma, es posible distinguir si los

correos descargados han sido vistos o no y procesarlos de acuerdo a ello.

Cada caso de uso implementado presenta pequeñas diferencias al interaccionar con el servidor IMAP. Como ejemplo, para la generación de estadísticas, la llamada *search* devuelve todos los identificadores del buzón, tanto leídos como no leídos. Para la obtención del histórico de una cuenta sólo se traen aquellos identificadores asociados a correos que están relacionados con la cuenta indicada.

La secuencia de peticiones *select-search-fetch* supone la asignación del *flag Seen* a todos los correos traídos del servidor. Esto no debería ser así, pues los correos no llegan a ser visualizados por el usuario final. Para mantener el estado inicial del buzón se utiliza la petición *store* a la cual se le proporciona los identificadores de los correos no vistos y el nuevo *flag*. A continuación, el cliente procesa los datos descargados de acuerdo al caso de uso específico y termina la sesión cerrando el buzón seleccionado mediante *close* y cierra la conexión con el servidor a través de *logout*.

La mayor parte del funcionamiento del sistema se produce entre las llamadas *fetch* y *store* que se aprecian en la Figura 6.1. Es aquí donde se introduce la lógica de los distintos casos de uso. Para la respuesta automática de correos la interacción con el módulo de clasificación y el servidor SMTP se produce íntegramente entre estas dos peticiones. En la Figura 6.2 se muestra la interacción con estos dos componentes.

Una vez se disponen de los correos en el cliente, se carga en memoria el modelo de clasificación de correos. Por cada correo con asunto *edoovillage* se llama a *predict*, al cual se le proporciona el cuerpo del mensaje y hace las transformaciones necesarias de éste para devolver una de las etiquetas.

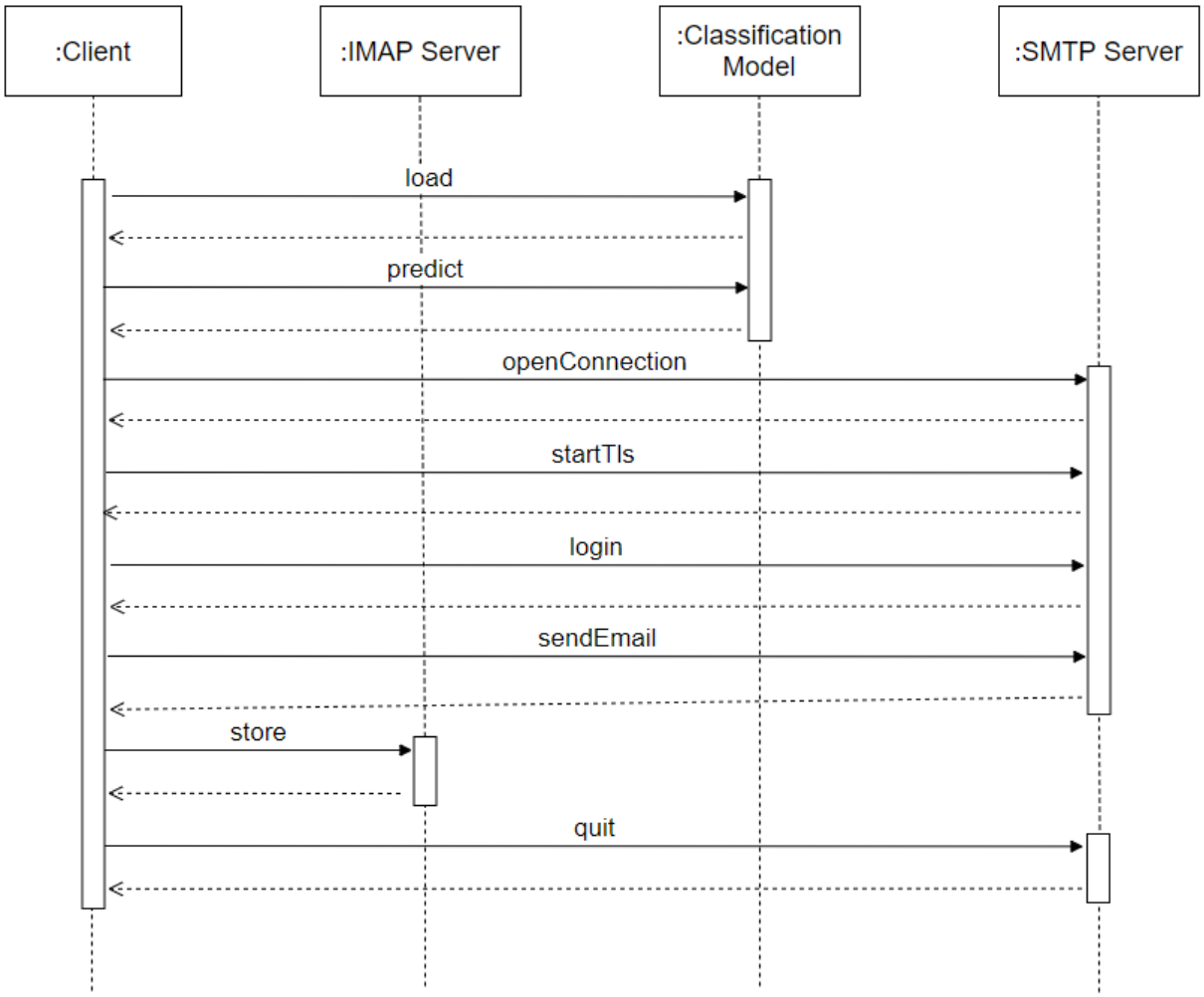


Figura 6.2: Diagrama de secuencia para respuesta automática de correos.

Cada correo a ser respondido, desencadena una sequencia de peticiones al servidor SMTP. *openConnection* abre una conexión con el servidor y *startTls* la pone en modo *TLS (Transport Layer Security)* cifrando los siguientes comandos. Después, se procede a la autenticación a través de *login*. Si ha ido correcto, se procede a enviar el email mediante una petición *sendMail* y a marcar éste como leído mediante una petición *store* al servidor IMAP. Finalmente, se termina la sesión SMTP y se cierra la conexión a través de *quit*.

6.2. Implementación

La implementación de la prueba de concepto se ha desarrollado en *Python* en un entorno *conda*. Para la interacción con los servidores IMAP y SMTP se han utilizado las librerías *imaplib* y *smtplib* respectivamente. El módulo de clasificación se ha desarrollado mediante *scikit-learn*, construyendo un *Pipeline* del proceso final que ha sido serializado utilizando *pickle*. Se ha utilizado *NLTK* para el procesamiento de los textos, *matplotlib* para la generación de gráficos y *email* para la maninulación de correos.

En la Figura 6.3 se muestra la estructura seguida para la implementación. Los casos de uso realizan llamadas a tres APIs diferentes. *API email* se encarga de gestionar la interacción con las buzones y correos del usuario, y de responder y enviar emails. *API data processing* se encarga de almacenar, transformar y realizar todas las modificaciones necesarias de los correos. Finalmente *API classification* se ocupa de la clasificación de emails y sólo es utilizada por el caso de uso de respuesta automática de correos.

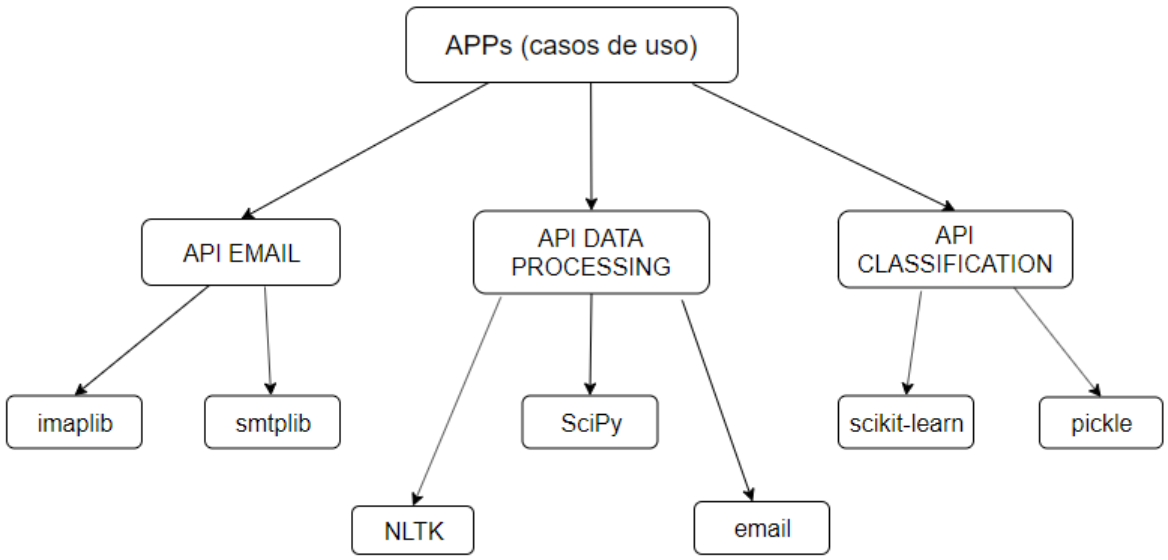


Figura 6.3: Estructura de la implementación.

Funciones de las APIs

A continuación se enumeran las funciones más relevantes para la interacción tanto con los servidores IMAP y SMTP, como de procesamiento de datos. Algunas no son necesarias para la prueba de concepto, como *delete_email* o *build_and_send_email*. Igualmente han sido implementadas para un futuro desarrollo del sistema:

- ***open_connection(hostname, username, password):*** Se encarga de establecer la conexión con el servidor IMAP sobre un socket SSL cifrado. Utiliza el puerto estándar (993) para las conexiones IMAP4 sobre SSL. Se le proporciona un *host* y las credenciales del usuario a autenticar. Devuelve un objeto *IMAP4_SSL* ya inicializado.

- ***close_connection(conn):*** Gestiona la finalización de la conexión con el servidor IMAP. Cierra el buzón seleccionado, los mensajes eliminados se quitan del buzón de escritura y termina la conexión con el servidor.

- ***retrieve_emails_by_flag(conn, flag, mailbox, preserve_unseen=True):*** Selecciona en el servidor IMAP el buzón indicado y valida el *flag* proporcionado. Se encarga de buscar los identificadores de los mensajes de acuerdo al *flag* y de descargarlos del servidor. En el caso de que *preserve_unseen* se mantenga a *True* los flags de los correos en el servidor no llegan a ser modificados. Devuelve un *DataFrame* que tiene por columnas lo campos: *uid, from, date, date_raw, subject, body* y *attachments*. A continuación se muestra la implementación realizada, destacando la llamada a *get_message_data()*.

```
def retrieve_emails_by_flag(conn, flag, mailbox, preserve_unseen=True,
verbose=False):
    conn.select(mailbox)    # Selecciona el buzón

    if flag not in ['Seen', 'Unseen', 'Answered', 'Unanswered']:
```

```
    return 'error'  # Valida el flag

# Busca correos en el buzón en función del flag
result_search, data_search = conn.uid('search', None, flag)

if result_search != 'OK':
    return 'error'  # Comprueba que la búsqueda tuvo éxito

# Decodifica los ids de los correos y los guarda como una lista
data_search = data_search[0].decode('utf-8').split()
# Inicializa el DataFrame para guardar los correos
database = pd.DataFrame(columns=['uid','from','date', 'date_raw',
'subject','body','attachments'])

for latest_email_uid in data_search:
    # Trae los datos del correo y los guarda como un diccionario
    email_dict = get_message_data(conn, latest_email_uid)

    # Convierte la fecha a formato YYYYMMDDHHMMSS
    raw_date = int(str(email_dict['Date']).replace(':','')
    .replace(' ', '').replace('-', ''))

    # Añade el correo al final de DataFrame
    database.loc[len(database)] = [str(latest_email_uid),
    email_dict['From'][1], email_dict['Date'], raw_date,
    email_dict['Subject'], email_dict['Body'], email_dict['Attachments']]
```

```
    if preserve_unseen:
        for x in data_search:
            # Marca los emails como no vistos
            set_flag_to_email(conn, x, 'Unseen')

    return database
```

- ***search_emails(conn, mailbox, field, value):*** Selecciona en el servidor IMAP el buzón indicado y trae los correos de acuerdo al campo y valor proporcionados. Posibles campos de búsqueda son: *from, to, body, date,* etc. Por cada uno de los identificadores encontrados se descargan los datos. Se devuelve un *DataFrame* que tiene por columnas los campos: uid, from, date, date_raw, subject, body, attachments.

- ***build_and_send_email(message_body, subject, recipient, username, password):*** Gestiona el envío de un nuevo mensaje dado los campos de éste. Resuelve toda la interacción que se realiza con el servidor SMTP.

- ***reply_email(conn, email_received, new_message_body, username, password):*** Dado un objeto de conexión *IMAP4_SSL*, el identificador del email recibido en el servidor, el cuerpo del mensaje de respuesta y las credenciales de autenticación, gestiona la respuesta del email en un mismo hilo. Utiliza el paquete *email* para construir la estructura completa del correo desde cero. Resuelve toda la interacción que se realiza con los servidores SMTP e IMAP (para poner el mensaje anterior como visto).

- ***set_flag_to_email(conn, uid, flag):*** Asigna *flag* al correo con identificador *uid*. Los *flag* aceptados son: *Seen, Answered, Flagged, Draft, Unseen, Unanswered.*

- ***delete_email(conn, uid):*** Dada una conexión *IMAP4_SSL* y el identificador de un correo, gestiona la eliminación de éste en el servidor.

- ***assign_recommended_actions_to_emails(database_unseen):*** El argumento de entrada es un *DataFrame* con columnas *subject*, *uid*, *from* y *body*. Se encarga de cargar el módulo de clasificación, realizar el preprocesamiento de los textos para proveerlos al modelo y de realizar la predicción. Devuelve el *DataFrame* de entrada con un campo *recommended_action* con el resultado de la predicción.

- ***get_message_data(conn, uid):*** Dado un objeto de conexión y el identificador de un correo, trae los datos de éste, los parsea y los devuelve como un diccionario.

- ***format_raw_email_mailparser(raw_email):*** Dado una secuencia de bytes correspondientes a la descarga de un correo del servidor IMAP, lo decodifica y parsea, y devuelve un email estructurado por los campos: *From, To, Date, Subject, Body, Attachments, Reply-To* y *Message-id*.

6.3. Experimentación

Clasificación

El proceso de desarrollo comenzó con la tarea de clasificación de los correos con asunto *edoovillage*. El dataset para realizar el entrenamiento presenta dos clases desbalanceadas:

- **solicitud** (solicitudes admitidas): 128 instancias.

- **not_fulfill_requirements** (solicitudes no admitidas): 32 instancias.

Dado el reducido tamaño del dataset, se ha optado por utilizar la traducción al inglés realizada en la Sección 5.2 para aquellos textos cuyo idioma original no fuese éste. De esta forma se evita realizar múltiples clasificadores por cada idioma, teniendo algunos de ellos un conjunto de entrenamiento muy reducido o realizar un clasificador con palabras de varios

idiomas aumentando considerablemente el número de *features*.

El procesamiento aplicado a los textos para suministrarlos a los algoritmos de clasificación incluye:

- Eliminación de caracteres especiales.
- Eliminación de caracteres sueltos.
- Conversión de los caracteres a minúscula.
- Lematización (WordNetLemmatizer).
- Eliminación de *stopwords*.
- Omisión de palabras que aparecen en más del 70 % de las instancias.

Debido al desbalanceo, se ha utilizado *random undersampling* en el entrenamiento para evitar que el clasificador asigne siempre la etiqueta más frecuente (*solicitud*). El conjunto de entrenamiento resultante presenta un número equivalente de instancias para ambas clases, igual al tamaño de la clase menos frecuente.

Para evaluar el clasificador se ha encontrado como principal problema el reducido tamaño del dataset, por lo que se ha optado por utilizar *leave-one-out cross validation*. Esta técnica entrena el clasificador N veces (con N el tamaño del dataset) con todas las instancias salvo una, que se utiliza para realizar la predicción. De esta forma se maximiza el número de instancias que se utiliza para entrenar el algoritmo permitiendo sacar métricas más representativas del modelo obtenido.

El modelo que mejor resultados ha arrojado se ha conseguido con el algoritmo *SGDClassifier* de la librería *scikit-learn*, el cual ajusta una SVM lineal. Los valores utilizados para

los parámetros son: *loss='log', penalty='l1', max_iter=5.*

En la tabla inferior se muestran las principales métricas obtenidas de la evaluación del clasificador. Se ha conseguido un alto nivel de acierto, alcanzando un *accuracy* del 87 %. No obstante hay diferencias notables entre los resultados obtenidos para las dos clases. La más frecuente, *solicitud*, supera el 90 % para las métricas de *precision* y *recall*. Esto no es así para la clase *no_cumple_requisitos* que se queda en un 71 % para *F1-score*.

	Precision	**Recall**	**F1-score**	**Support**
no_cumple_requisitos	0.67	0.75	0.71	32
solicitud	0.94	0.91	0.92	128
Accuracy			0.87	160

El objetivo principal es reducir los falsos negativos, de manera que solicitantes aptos no sean rechazados. Este dato podemos obtenerlo de la métrica *recall* para la clase *solicitud*, la cual indica que de aquellas solicitudes que deberían ser aceptadas, se reconocen el 91 %.

Casos de uso

Cada caso de uso se ha desarrollado mediante un *script*, que realiza llamadas a las APIs para implementarlo. Cada uno de ellos proporcionan toda la funcionalidad definida en la Sección 4.3. A continuación se muestran los resultados obtenidos.

Respuesta automática de correos

Para mostrar como el caso de uso queda cubierto por la implementación, en la Figura 6.4 se presenta el resultado de una ejecución de prueba. Por claridad, y de manera que no se repitan llamadas como *get_message_data()* la prueba se realiza para una bandeja de entrada con un sólo correo a responder. Se marcan en verde las llamadas principales a las

APIs, que a su vez realizan llamadas a otras funciones dentro de éstas.

```
Enter your email username --> tfgucm2019@gmail.com
Enter your password -->
open_connection()
## Connecting to imap.gmail.com
## Logging in as tfgucm2019@gmail.com
retrieve_emails_by_flag()
## Selecting inbox mailbox
## Retrieving emails from inbox with flag Unseen
## Searching for matching emails
## get_message_data()
#### Fetching message data
#### format_raw_email_mailparser()
## set_flag_to_email()
#### Storing unseen flag for email
assign_recommended_actions_to_emails()
## Loading classification model
## text_processing()
## Parsing emails by subject
## Applying feature engineering and predicting recommended actions
- Classification report:
- not_fulfill_requirements    1
Filtering automatable emails
reply_email()
## Creating email object and setting params
## send_email()
#### Starting connection
#### Loggin in as tfgucm2019@gmail.com
#### Sending email
#### Shutting down connection
## set_flag_to_email()
#### Storing seen flag for email
- 1 of 1 emails were replayed
close_connection()
## Shutting down connection with server
## Successfully logged out
```

Figura 6.4: Ejemplo de la sequencia de llamadas y tareas para la respuesta automática.

La ejecución comienza solicitando al usuario las credenciales de su cuenta. Esto es lo único que tendrá que hacer para que la ejecución se lleve a cabo con éxito. En el caso de no facilitar una dirección de email válida se vuelve a pedir una cuenta de correo. A continuación se llama a *open_ connection()* que abrirá una conexión con el servidor IMAP y logeará al usuario.

A continuación, se traen los mensajes no vistos que el usuario tiene en su buzón de entrada. Esto se realiza a través de *retrieve_ emails_ by_ flag()*, que lleva a cabo la interacción

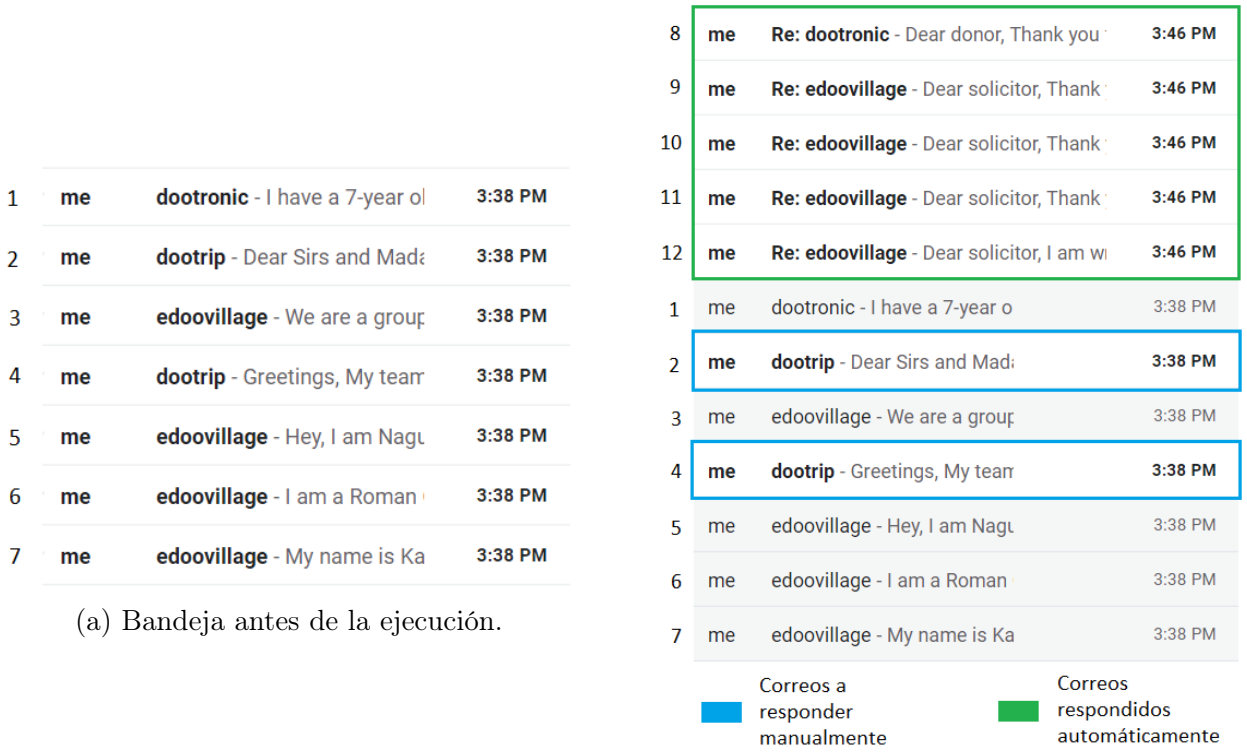
con el servidor y llama a *get_ message_ data()* por cada uno de los correos. Esta función trae los correos, los decodifica y parsea para que sea más fácil trabajar con ellos. Este proceso supone marcar los emails como vistos. Esto se corrige mediante *set_ flag_ to_ email()*, que se llama una vez por cada correo que hayamos traído del servidor.

Seguidamente, se asignan las acciones recomendadas a los emails. Se comienza por cargar el modelo de clasificación y procesar los textos de los mensajes de manera que puedan ser proporcionados al modelo. El parseo del asunto de los mensajes determina aquellos correos que serán provistos al clasificador. Sólo a aquéllos correspondientes a *edoovillage* se les aplicarán las transformaciones necesarias para la clasificación. En la Figura 6.4 se observa como en la bandeja de entrada hay un único correo perteneciente a una solicitud y que el clasificador etiqueta (classification report) como que no cumple los requisitos necesarios.

A continuación se filtran aquellos emails que pertenecen a categorías que deben responderse manualmente, y se procede a responder aquellos pertenecientes a las categorías que sí. Por cada email a contestar se ejecutará la función *reply_ email* la cual crea el objeto acorde y lo envía mediante *send_ email()*, que se encarga de la interacción presentada en la Sección 6.1. Finalmente se procede a cerrar la conexión con el servidor.

La Figura 6.5 muestra el resultado de la respuesta automática de emails. Se ha decidido trabajar con una sóla cuenta de correo de manera que todos los resultados puedan observarse en una misma bandeja de entrada y resulte más fácil la comprensión del caso.

En 6.5a se observa el estado de la bandeja antes de la ejecución. Podemos apreciar como hay correos entrantes de tres tipos mirando el asunto. Si examinamos estos mismos correos en 6.5b, después de la ejecución, vemos como aquellos pertenecientes a *dootrip* (en azul) siguen como vistos, pues no han sido respondidos, y son los voluntarios los que deben revi-



(a) Bandeja antes de la ejecución.

(b) Bandeja tras la ejecución.

Figura 6.5: Resultados de la respuesta automática de correos.

sarlos.

Para los correos con asunto *dootronic* o *edoovillage*, vemos como han sido marcados como vistos y como por cada uno de ellos aparece un nuevo email de entrada (en verde), marcado como no visto. Además se observa la inclusión de *Re:* al comienzo del asunto y el cuerpo del mensaje asignado a cada correo en función éste. De aquellos emails con asunto *edoovillage* que han sido respondidos, destacar el que tiene un cuerpo distinto, que se ha clasificado como que no cumple los requisitos.

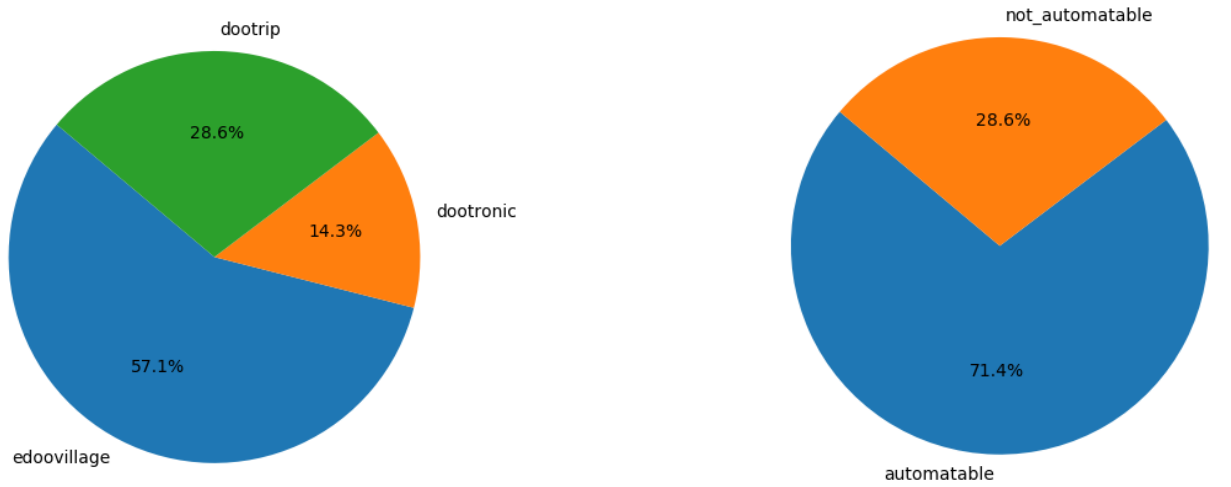
Recuperación del histórico de cuenta

Este caso de uso tiene un comportamiento similar al campo de búsqueda de muchos clientes de correo electrónico, siendo el valor de búsqueda una cuenta. Este caso se plantea

de cara a implementar un cliente específico con todas las funcionalidades básicas.

Generación de estadísticas generales

Este caso de uso genera estadísticas en forma de diagramas con el objetivo de ser integrados en una interfaz gráfica en el futuro. En la Figura 6.6 se muestran ejemplos de distribuciones generadas para la bandeja de entrada de la Figura 6.5a. El Diagrama 6.6a muestra las distintas proporciones de los correos por asunto. Dado el reducido tamaño de la bandeja de entrada, es posible apreciar la correspondencia entre las porciones y el número de emails por asunto. El Diagrama 6.6b, muestra la proporción de correos que podrían ser respondidos. En nuestro caso, aquellos pertenecientes a *dootrip* son los únicos que hay que contestar obligadamente a mano y de ahí la correspondencia en la porción en los dos diagramas.



(a) Distribución de emails por asunto.

(b) Distribución de emails por automatización.

Figura 6.6: Distribuciones de la bandeja de entrada.

Para mostrar como todo el caso de uso queda resuelto, se ha realizado una prueba para la bandeja de entrada de la Figura 6.7. Ésta se ha mantenido sencilla para facilitar la com-

prensión, y a la vez variada, de forma que se pueda percibir toda la casuística. Se incluye un correo leído en la bandeja, que no debería ser tenido en cuenta para la generación de estadísticas. Se han utilizado dos correos para enviar los mensajes de prueba, *julareña@ucm.es* y *juliquiag@hotmail.com*.

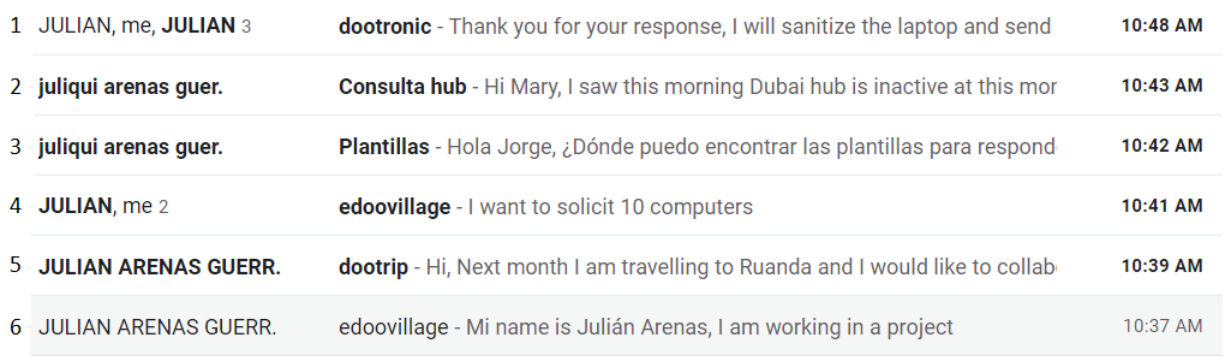
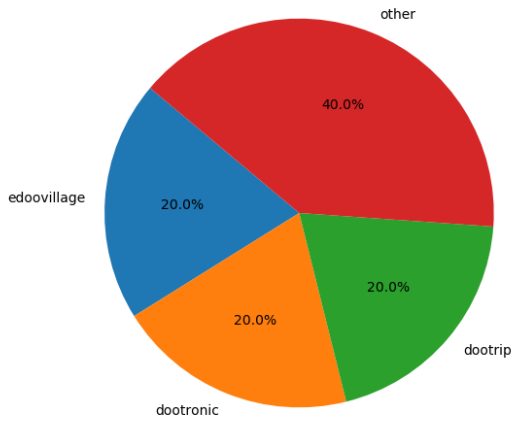


Figura 6.7: Bandeja de entrada de prueba.

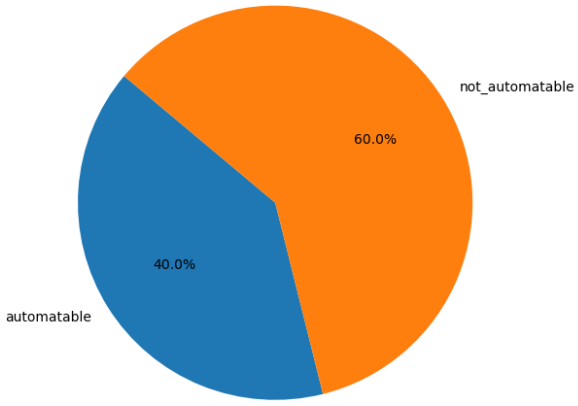
En la Figura 6.8 se presentan las mismas gráficas que anteriormente. Se vuelve a comprobar la correspondencia de porcentajes de los asuntos entre la nueva bandeja de entrada y el Diagrama 6.8a, y como la porción automatizable en la Figura 6.8b concuerda con las de *edoovillage* y *doootronic*.

La Figura 6.10a representa el balance de correos internos y externos. Como no se tiene una cuenta de Labdoo para realizar pruebas, se consideran como internos aquellos correos provenientes del dominio *hotmail.com*. En concreto los correos con asuntos *Consulta hub* y *Plantillas* provienen de una cuenta con este dominio, es decir un 40 % de los correos de la bandeja de entrada.

La tipología de los correos de la bandeja se muestra en 6.10b. La más inmediata es la de nuevos *issues*, que queda representada por aquellos correos cuyos asuntos pertenecen a algunas de las categorías de la Sección 4.2 y que no forman parte de un hilo. Los *issues*

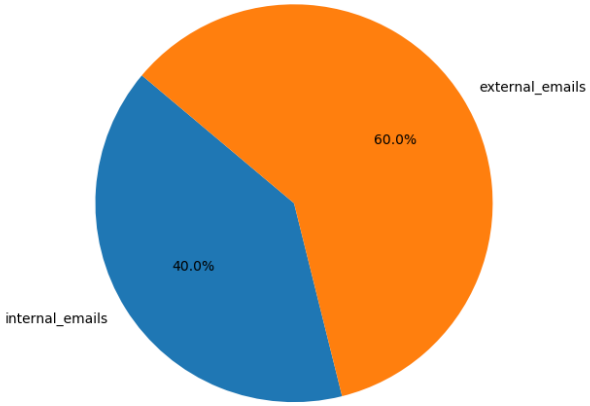


(a) Distribución de emails por asunto.

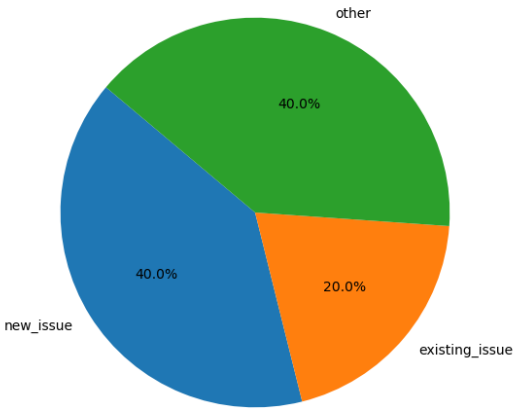


(b) Distribución de emails por automatización.

Figura 6.8: Estadísticas para bandeja de entrada.



(a) Distribución de emails interno y externos.



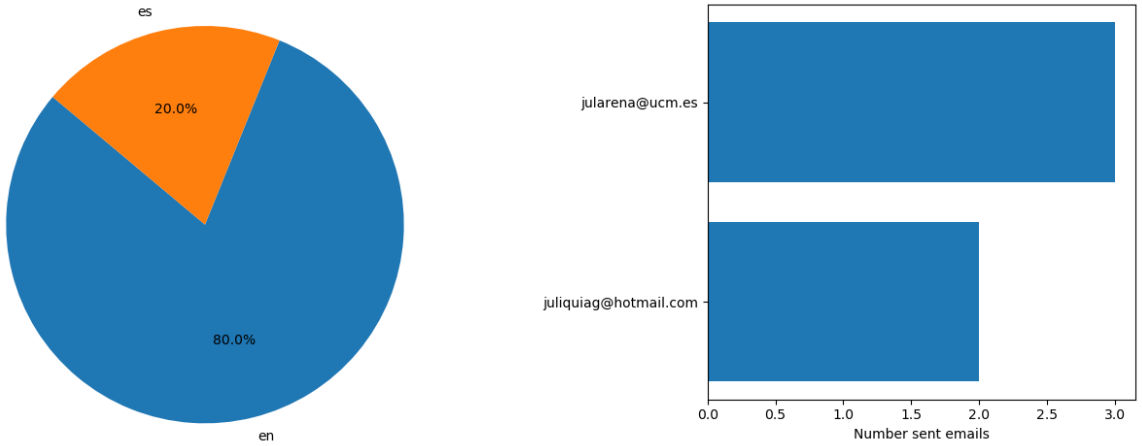
(b) Distribución del tipo de emails.

Figura 6.9: Estadísticas para bandeja de entrada.

existentes quedan representados por correos con asuntos en algunas de esas categorías y que forman parte de un hilo, en nuestra prueba el email con asunto *dootronic*. Finalmente, el resto de correos que no cumple estas condiciones se clasifican como otros, que en nuestro caso coinciden con los correos internos.

En la Figura 6.10a se aprecia la identificación de idiomas realizada. Sólo uno de los mensajes está escrito en castellano, y el resto en inglés. La Figura 6.10b muestra un ranking con

el número de correos que tenemos en la bandeja de entrada por usuario. La bandeja de la prueba solo tiene emails de dos usuarios, uno representando un compañero de la organización (el que queda más abajo), y otro una persona externa.



(a) Distribución de emails por idiomas.

(b) Ranking de remitentes.

Figura 6.10: Estadísticas para bandeja de entrada.

Si bien es posible realizar el caso de uso de recuperación de histórico mediante cualquier cliente de correo electrónico actual, esto no es así para los otros dos casos implementados. Esto supone que la prueba de concepto realizada, aun no siendo una herramienta final, podría empezar a utilizarse como complemento a un cliente de correo por el personal de Labdoo.

Capítulo 7

Conclusiones y trabajo futuro

En este trabajo se ha analizado la problemática de la **gestión de correo electrónico** en el caso de las ONGs. El propósito principal del proyecto es minimizar el problema de *email overload* en este tipo de organizaciones a través de la **automatización** y facilitación de tareas.

Con este objetivo, se han estudiado los últimos avances en *workflow management*, y el papel que el correo electrónico desempeña en estos sistemas y en las organizaciones actuales. Además, se ha llevado a cabo un análisis de las **necesidades** particulares de las ONGs y sus limitaciones, ligadas a su carácter no lucrativo, lo que ha permitido establecer una serie de requisitos para desarrollar el sistema.

Además, se ha llevado a cabo un análisis de **datos** proporcionados por Labdoo, que se han preprocesado y etiquetado de cara a entrenar algoritmos de aprendizaje supervisado. Esto ha ayudado a tener un conocimiento más profundo de la problemática concreta a la que se enfrenta la organización y proponer una solución basada en datos.

Teniendo presente todo lo anterior, se ha desarrollado una **prueba de concepto** para el caso de Labdoo que contempla los tres casos de uso de mayor prioridad. La funcionalidad

más útil es la **clasificación y respuesta automática de emails**, que reduce drásticamente la carga de trabajo asociada a la gestión del correo. En el estado actual de desarrollo, el 85 % de los emails pueden tratarse de manera automática y, dentro de estos, el número de falsos negativos ha quedado reducido por debajo del 10 %, tasa de fallo asumible por un sistema productivizado.

Por otra parte, la **generación de estadísticas** y la **recuperación del histórico** de una cuenta facilitan las tareas de los usuarios y les ayudan a definir estrategias para abordar el buzón de entrada. Así, el sistema propuesto ofrece **diversas funcionalidades** que asisten a los/as voluntarios/as en sus tareas de manera efectiva, objetivo principal del presente trabajo.

Además, es de destacar que la automatización de tareas sencillas y abordables por la Inteligencia Artificial permite la adaptación a las necesidades específicas de la organización y el **perfeccionamiento** progresivo de los resultados, por lo que el margen de mejora es amplio. Esto constituye una de las principales ventajas del sistema. En vista a todo lo anterior, los resultados de las prueba de concepto se consideran muy prometedores.

La principal limitación de la propuesta está relacionada con el reducido tamaño del dataset, que restringe el rendimiento de la clasificación. Por lo tanto, el **trabajo futuro** debe orientarse hacia la recavación de datos de entrenamiento más amplios, de manera que los algoritmos puedan generalizar mejor el problema, y aumentar la tasa de acierto.

Además, tras un determinado tiempo de uso del prototipo desarrollado, sería beneficioso contar con *feedback* en forma de comentarios y opiniones del personal, de cara a incluir nuevas mejoras y avanzar en la eficiencia. En un trabajo más amplio, el uso de datos de otras organizaciones similares a Labdoo permitiría generalizar la propuesta y desarrollar

una implementación más diversa.

Por último, se plantea como trabajo futuro el desarrollo de una interfaz gráfica amigable, que haga uso de la API desarrollada. Esto facilitaría su uso en el día a día del personal de la ONG, y disminuiría el tiempo de aprendizaje de los/as nuevos/as voluntarios/as, incidiendo de manera positiva en el funcionamiento de la organización.

En conclusión, el trabajo realizado constituye un **punto de partida** para la mejora de la gestión del correo electrónico en organizaciones de escasos recursos. Ésta es una problemática extendida, aunque ha sido en general poco explorada. Las conclusiones generales de este trabajo muestran que **la automatización es viable y los resultados son satisfactorios**, si bien es necesario avanzar en el futuro en la propuesta para desarrollar una solución óptima del problema, aplicable a distintas organizaciones.

Capítulo 8

Conclusions and future work

This work has analyzed the problem of **email management** in the case of NGOs. The main purpose of the project is to minimize the problem of *email overload* in this type of organizations through the **automation** and facilitation of tasks.

With this objective, we have studied the latest advances in workflow management, and the role that email plays in these systems and in current organizations. In addition, an analysis of the particular **needs** of NGOs and their limitations, linked to their nonprofit nature, has been carried out, which has allowed establishing a series of requirements to develop the system.

With all of the above in mind, a **proof of concept** has been developed for the Labdoo case that considers the three highest priority use cases. The most useful functionality is the classification and automatic response of emails, that drastically reduces the workload associated with email management. In the current state of development, 85 % of emails can be treated automatically and, within these, the number of false negatives has been reduced below 10 %, a failure rate acceptable by a productivized system.

On the other hand, the **generation of statistics** and the recovery of the archive of

an account facilitate the tasks of the users and help them define strategies to address the inbox. Thus, the proposed system offers **various functionalities** that assist volunteers in their tasks effectively, main objective of this work.

In addition, it is noteworthy that the automation of simple and approachable tasks by Artificial Intelligence allows adaptation to the specific needs of the organization and the progressive **improvement** of the results, so that the scope for upgrading is wide. This constitutes one of the main advantages of the system. In view of the above, the results of the proof of concept are considered very promising.

The main limitation of the proposal is related to the small size of the dataset, which restricts the performance of the classification. Therefore, **future work** should be geared towards gathering more extensive training data, so that algorithms can better generalize the problem, and increase the success rate.

In addition, after a certain period of use of the developed prototype, it would be beneficial to have feedback in the form of comments and opinions of the staff, in order to include new improvements and gain efficiency. In a broader work, the use of data from other organizations similar to Labdoo would allow generalizing the proposal and elaborate a more diverse implementation.

Finally, the development of a friendly graphical interface that makes use of the designed API is considered as future work. This would facilitate its day-to-day use by NGO staff, and reduce the learning time of new volunteers, positively affecting the operation of the organization.

In conclusion, the work done constitutes a **starting point** for the improvement of email

management in low-income organizations. This is an extended problem, although in general it has been little explored. The general conclusions of this work show that **automation is viable and the results are satisfactory**, although, in the future, it is necessary to keep advancing in the proposal in order to develop an optimal solution to the problem, applicable to different organizations.

Bibliografía

[1] About labdoo. https://www.labdoo.org/content/about-labdoo, 2019.

[2] Cloud translation documentation. https://cloud.google.com/translate/docs/, 2019.

[3] F1 score. https://en.wikipedia.org/wiki/F1_score, 2019.

[4] The labdoo social network - how it works. https://www.labdoo.org/book/export/html/39, 2019.

[5] Modelo bolsa de palabras. https://es.wikipedia.org/wiki/Modelo_bolsa_de_palabras, 2019.

[6] Text classification a comprehensive guide to classifying text with machine learning. https://monkeylearn.com/text-classification/, 2019.

[7] What is inbox zero. https://flow-e.com/inbox-zero, 2019.

[8] Workflow management system. https://en.wikipedia.org/wiki/Workflow_management_system, 2019.

[9] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, and Ian Smith. Taskmaster: Recasting email as task management. 01 2002.

[10] Victoria Bellotti, Nicolas Ducheneaut, Mark Howard, and Ian Smith. Taking email to task: the design and evaluation of a task management centered email tool. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 345–352. ACM, 2003.

[11] Shefali Bhat, C Anantaram, and Hemant K Jain. An architecture for intelligent email based workflow interface to business applications. In *IC-AI*, pages 344–350, 2008.

[12] Thomas Burkhart, Dirk Werth, and Peter Loos. Context-sensitive business process support based on emails. In *Proceedings of the 21st International Conference on World Wide Web*, pages 851–856. ACM, 2012.

[13] Laura A Dabbish and Robert E Kraut. Email overload at work: an analysis of factors associated with email strain. In *Proceedings of the 2006 20th anniversary conference on Computer supported cooperative work*, pages 431–440. ACM, 2006.

[14] Nicolas Ducheneaut and Victoria Bellotti. E-mail as habitat: an exploration of embedded personal information management. *interactions*, 8(5):30–38, 2001.

[15] Andrew Faulring, Brad Myers, Ken Mohnkern, Bradley Schmerl, Aaron Steinfeld, John Zimmerman, Asim Smailagic, Jeffery Hansen, and Daniel Siewiorek. Agent-assisted task management that reduces email overload. In *Proceedings of the 15th international conference on Intelligent user interfaces*, pages 61–70. ACM, 2010.

[16] Jorge Franganillo. El correo electrónico en las organizaciones: Retos, oportunidades y tendencias en la gestión de información personal. *Comunicación corporativa en Red.*

[17] Carolin Gomulia. Improving coordination, integration and communication in ngos. http://www.ngopulse.org/article/2015/08/05/improving-coordination-integration-and-communication-ngos-some-practical, 2015.

[18] Thomas Jackson, Ray Dawson, and Darren Wilson. The cost of email interruption. *Journal of Systems and Information Technology*, 5, 04 2004.

[19] Zukhanye N Kwinana, P Wentworth, and A Terzoli. An email based issue-tracking workflow system that is extensible across organizational boundaries. SATNAC, 2004.

[20] Kim McMurtry. Managing email overload in the workplace. *Performance Improvement*, 53(7):31–37, 2014.

[21] Ghulam Mujtaba, Liyana Shuib, Ram Gopal Raj, Nahdia Majeed, and Mohammed Ali Al-Garadi. Email classification research trends: Review and open issues. *IEEE Access*, 5:9044–9064, 2017.

[22] Chun Ouyang, Michael Adams, Moe Thandar Wynn, and Arthur HM ter Hofstede. Workflow management. In *Handbook on Business Process Management 1*, pages 475–506. Springer, 2015.

[23] Jorge J. Gómez Sanz. A python program that takes an mbox file and anonymizes it partially. https://github.com/escalope/mboxanonymizer, 2019.

[24] Jakub Swacha et al. Management by email reinterpreted with a process-based approach. *Managing Intellectual Capital and Innovation for Sustainable and Inclusive Society: Managing Intellectual Capital and Innovation*, pages 1343–1350, 2015.

[25] Wil MP Van der Aalst. The application of petri nets to workflow management. *Journal of circuits, systems, and computers*, 8(01):21–66, 1998.